Nintendo® FAMILY COMPUTER™

—— ファミリー ベーシック™で遊ぼう! ——

ぼくとわたしの
ファミリーベーシック™教室



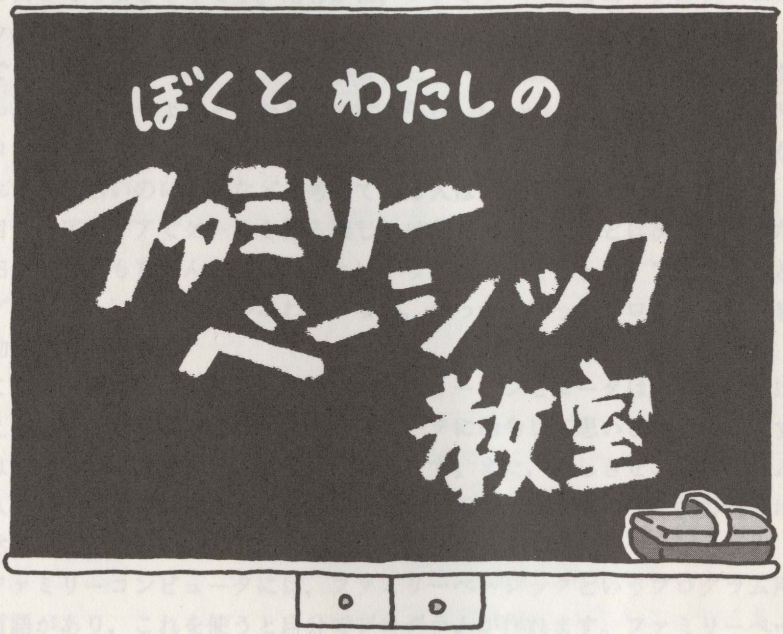
特典がいっぱい!
会員募集中!!
ファミコン
ファンクラブ
会員申し込み用紙付き

マイクロデザイン編

販売元 マイクロハウス

Nintendo® FAMILY COMPUTER™

—ファミリー ベーシック™で遊ぼう!—



※ファミリー コンピュータ™、ファミリー ベーシック™は、任天堂の商標です。

はじめに

ファミリーコンピュータで遊ぶゲームの面白さは，本体（コンソール）が300万台以上も販売されていることを考えると，もう日本国中に知れ渡っていると言ってもよさそうです。たしかに，ファミリーコンピュータのゲームカセットは，よくできていて楽しめます。

でも，ファミリーコンピュータがコンピュータであるのなら，もっと別な使い方があってもよさそうなものです。『パソコンとおなじように，ファミリーコンピュータでも自作のゲームで遊んでみたい！』，『自分でプログラムが作れたら面白いのに！』などと考えている人はいませんか？

自分のアイデアでゲームを作る楽しみは，ただ遊ぶだけとは次元の違った面白さです。もちろん，楽しいことばかりではないでしょう。アイデアをどうプログラムにしたらいいかわからない，せっかく作ったプログラムがうまく動かない……etc.

これは，あたりまえのことです。ファミリーコンピュータは，小さいとはいえ立派なコンピュータです。コンピュータに命令して思いどおりに動かすには，残念ながら多少の知識は必要なのです。また，おしきせのプログラムを入力して遊ぶのではなく，自分で作り出すのですから少しぐらいの苦労は当然です。

ファミリーコンピュータには，ファミリーベーシックというプログラム用の言語があり，これを使うと自分でプログラムが作れます。ファミリーベーシックは，ゲームを作るために開発されたベーシックですから，ゲームを作るのが得意です。でも，この本では，あえて基本的なベーシック命令だけを取り上げました。というのは，先に『ファミリーベーシックで遊ぼう！』を発行した際に寄せられた，読者からの質問をヒントに，プログラムを自分で考え，自分で作れるようになることを目標にしたからです。

本書により，一人でも多くのちびっこプログラマが誕生することを望んでやみません。

編著者しるす

もくじ

第1章　ベーシックってこんなもの

第2章　こうして動かすベーシック

もくじ

第3章　基本コマンドでOK！

第4章　ゲームライブラリー

表紙・イラスト　佐々木正樹

第1章 ベーシックってこんなもの



1 ベーシックって何だ？

ファミリーコンピュータで，ベースボールやＦ１レースを楽しんでいるキミたち。ファミリーコンピュータがむずかしいものだとは，これっぽっちも思っていないでしょう。それどころか，スイッチを切るのがおしいくらいじゃないですか？

それでは，ファミリーベーシックのほうはどうかな？「これはちょっとむずかしいぞ」と思っているのではないですか？

同じファミリーコンピュータなのに，どうしてゲームカセットで遊んでいるときは楽しくて，ファミリーベーシックになるとむずかしくなるのでしょう？──それは，ゲームカセットだとすぐに遊べるけれど，ファミリーベーシックでは，まず自分でゲームを作ってからでないと遊べないからではないかと思います。だから，ファミリーベーシックには，ミュージックボードやバイオリズム占いが付いているし，ファミリーベーシックＶ３（以下Ｖ３と略します）には，すぐに遊べるようにゲームが入っているのです。

でも，ちょっと考えてみましょう。たしかに，すぐ遊べるほうが楽しいに決ってるけれど，自分でゲームを作るっていうのは，これはスゴイことじゃないかな？　ほかの誰かが作ったゲームで遊ぶより，自分で作ったゲームで遊ぶほうが何倍もオモシロイし，だいいち「高級」だと思いませんか？

ところで，ベーシックっていうのは，ファミリーコンピュータに命令して，

テレビに色を出したり，音を出したりするための言葉なんです。いままでキミたちが遊んでいたゲームカセットも，マシン語という別な言葉でできている，命令語（コマンド）のカタマリなんですよ。命令語のカタマリのことを，ふつう「プログラム」って呼んでいますが，このマシン語でプログラムを作るには，何年も勉強しないとチョット無理だと思います。

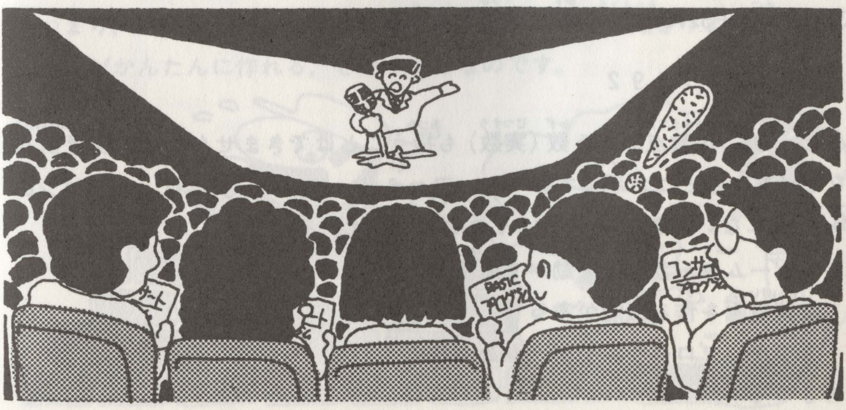
けれども，ベーシックなら，キミたちでもこれからすぐにプログラムが作れるかもしれません。だから，これからキミたちにプログラムの作り方を説明しましょう。

ところで，プログラムといったって，別にむずかしく考えることはないのです。コンサートや映画のプログラムと同じで，“コレの次はアレをして，その次はコレ……”というように，“何をするか”を順序よくならべたものだと思えば間違いはありません。

ファミリーコンピュータが“何をするか”は，ベーシック命令と呼ばれている“決まり文句”で違ってくるのです。だから，この文句（英語に近いので，なんだか呪文みたいだナと思うけど）をたくさん覚えると，いろいろなことをファミリーコンピュータにさせることができます。たとえば，マリオをテレビに出して歩かせたり，宇宙船が飛び回ったり……というようにです。

でも，いきなり自分の思いどおりに動かそうとしても，それはチョット無理というもの。やっぱり基礎から覚えていかないと……。

だから，この本ではあまりたくさんの命令を欲ばらないで，これだけはゼッタイ覚えておいてほしい命令だけを説明することにしましょう。



2 パソコンのベーシックとファミリーベーシックのちがい

ここで，ふつうのパソコンに内蔵されているBASICと，ファミリーベーシックとのちがいについて説明しておきましょう。

両方ともBASICであることに間違いはないのですが，実は大変な違いがあります。それは，この2つのBASICは，作られた目的がぜんぜん異なっている，ということなのです。

パソコンのBASICは，いろいろなところで使われることを考えて，たくさんの命令が盛り込まれています。計算もちゃんとこなせるし，グラフを描くこともできます。ですから，パソコンのBASICなら実用的な目的で，プログラムを組むことができるわけです。

ところが，ファミリーベーシックはそうではありません。ファミリーベーシックは，GAME BASICと呼ばれているとおり，ゲームを作るために生まれてきたBASICなのです。もちろん，ほかのことがぜんぜんできないわけではありませんが，やっぱりゲームの方が得意なのです。

たとえば，ふつうのパソコンでは，

845210000

などという大きな数を扱うことができますが，ファミリーベーシックでは，−32768から＋32767までの数しか使えません。また，

3．141592

のような，小数点のある数（実数）も扱うことはできませんから，技術計算などはちょっとニガ手です。でも，ファミリーベーシックには，パソコンのBASICにない大きな特長がいくつかあります。それは，

①ゲームキャラクタを動かす命令がある
②背景画を作る機能がある
③短かいプログラムでゲームを作れる

の3つです。

ゲームキャラクタとは，マリオやレディやペンペンなど，あらかじめ決められた16種類の動きのあるパターンのことです。それぞれのゲームキャラクタは，そのキャラクタの性格によって，歩いたり，飛んでいたりするかのように見えるパターンが用意されていて，かんたんなプログラムで動かせるようになっています。

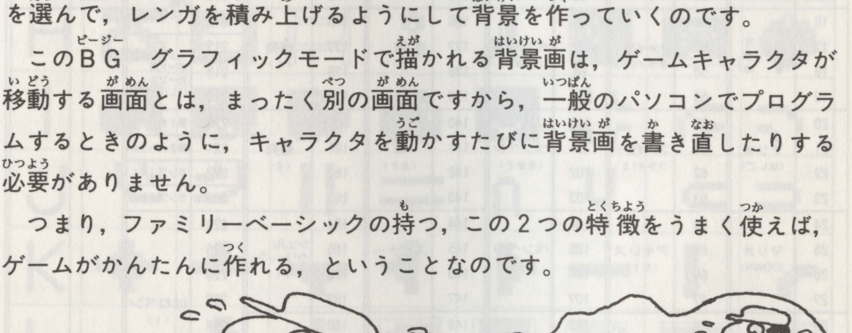
一般のパソコンで，1つのパターンを画面上で動かしたり，歩いているような動きを作ることは，なかなか大変な作業です。おそらくBASICのプログラムだけではできないでしょう。また，できたとしても，動くスピードが遅く，ゲームなどには使えません。

ところが，ファミリーベーシックでは，こうしたプログラムがとてもかんたんに作れて，しかも，びっくりするほど速くキャラクタを動かすことができます。

また，背景画は，ＢＧ　グラフィックモード（Background　GRAPHICの略）を使って，画面を672個に分け（28×24）その1つ1つの部分に，104種のパターンの中から選んで表示させることができます。また，そのパターンは4種類の配色の中から選んで色を変えることができます。つまり，パーツを選んで，レンガを積み上げるようにして背景を作っていくのです。

このＢＧ　グラフィックモードで描かれる背景画は，ゲームキャラクタが移動する画面とは，まったく別の画面ですから，一般のパソコンでプログラムするときのように，キャラクタを動かすたびに背景画を書き直したりする必要がありません。

つまり，ファミリーベーシックの持つ，この2つの特徴をうまく使えば，ゲームがかんたんに作れる，ということなのです。



★キャラクタテーブルA

ファミリーベーシックのキャラクタコードを，一覧表で示します。ＢＡＳＩＣのＣＨＲ＄（n），ＡＳＣ（"A$"）による文字とコードの変換は，これに対応しています。キャラクタコード表A，またはBのキャラクタ，文字，記号は，バックグラウンド，スプライトの各面で使用できます。（どの面にどちらの表を使用するかの指定はＣＧＥＮ命令を使用します）

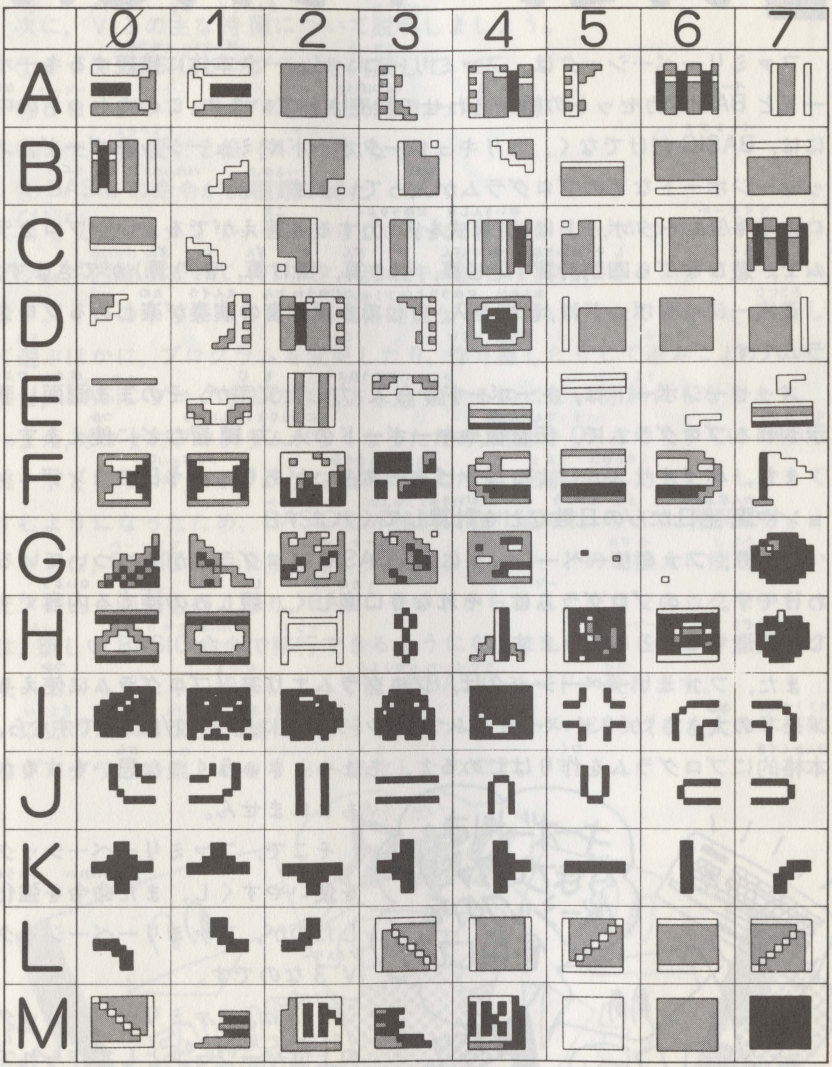
●キャラクタコード表Ａ（主に，スプライト用）

コード（10進数）は，キャラクタテーブルＡでアニメキャラクタの四隅に表記してある番号に対応します。

- アニメキャラクタを定義する場合は，このコード表にあるコード(10進数，または16進数)で指定してください。
- 0～255までの10進と16進の換算表としてもご利用ください。

コード (10進数)	説明	コード (10進数)	説明	コード (10進数)	説明	コード (10進数)	説明	コード (10進数)	説明	コード (10進数)	説明	コード (10進数)	説明
0		40		80		120		160		200		240	1
1	マリオ (WALK1)	41	レディ (JUMP)	81	アキレス (上1)	121	車 (左1)	161	スターキラー (上)	201	ニットピッカー (1)	241	2
2		42		82		122		162		202		242	3
3		43		83		123		163		203		243	4
4		44		84		124		164		204		244	COMPUTERの文字
5	マリオ (WALK2)	45	レディ (スリップ着地)	85	アキレス (上2)	125	車 (左2)	165	スターシップ (左)	205	ニットピッカー (2)	245	
6		46		86		126		166		206		246	
7		47		87		127		167		207		247	
8		48		88		128		168		208	レーザー (よこ)	248	
9	マリオ (WALK3)	49	レディ (はしご)	89	ニタニタ (1)	129	車 (左上1)	169	スターシップ (左上)	209		249	
10		50		90		130		170		210	レーザー (ななめ)	250	OPERATORの文字
11		51		91		131		171		211		251	
12		52		92		132		172		212	レーザー (たて)	252	
13	マリオ (JUMP)	53	レディ (DOWN)	93	ニタニタ (2)	133	車 (左上2)	173	スターシップ (上)	213		253	
14		54		94		134		174		214	スプライト用ぬりつぶしの色 (1),(2),(3)	254	
15		55		95		135		175		215		255	
16		56		96		136		176		216			
17	マリオ (スリップ着地)	57	ファイターフライ(1)	97	ペンペン (左歩1)	137	車 (上1)	177	爆発 (1)	217	ミュージックボード用カーソル (1),(2),(3)		
18		58		98		138		178		218			
19		59		99		139		179		219			
20		60		100		140		180		2200	第1カーソル		
21	マリオ (はしご)	61	ファイターフライ(2)	101	ペンペン (左歩2)	141	車 (上2)	181	爆発 (2)	221	第2カーソル		
22		62		102		142		182		222	ランプ (1)		
23		63		103		143		183		223	ランプ (2)		
24		64		104		144		184		224			
25	マリオ (DOWN)	65	アキレス (左1)	105	ペンペン (正面)	145	スピナー (1)	185	シェルクリーパー (1)	225			
26		66		106		146		186		226			
27		67		107		147		187		227	はねペン (1)		
28		68		108		148		188		228			
29	レディ (WALK1)	69	アキレス (左2)	109	ペンペン (後)	149	スピナー (2)	189	シェルクリーパー (2)	229			
30		70		110		150		190		230			
31		71		111		151		191		231			
32		72		112		152		192		232			
33	レディ (WALK2)	73	アキレス (左上1)	113	ファイアーボール(1)	153	スターキラー (左)	193	サイドステッパー (1)	233			
34		74		114		154		194		234			
35		75		115		155		195		235	はねペン (2)		
36		76		116		156		196		236			
37	レディ (WALK3)	77	アキレス (左上2)	117	ファイアーボール(2)	157	スターキラー (左上)	197	サイドステッパー (2)	237			
38		78		118		158		198		238			
39		79		119		159		199		239			

★キャラクタテーブルB



3 ファミリーベーシックとV3

ファミリーベーシックは，ファミリーコンピュータ本体に接続するキーボードとBASICカセットの組み合わせで発売されています。このカセットの中には，BASICだけでなく，カリキュレータボードやミュージックボード，メッセージボードなどのプログラムが入っています。

カリキュレータボードは，計算式を入力すると答えがでるというプログラムで，遊びながら四則計算（足し算，引き算，掛け算，割り算）ができます。

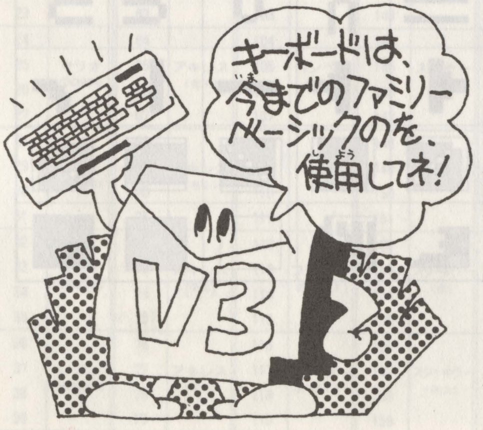
ミュージックボードは，音階を入力して多重和音の演奏が楽しめるプログラムです。

メッセージボードは，キーボードから入力した文字が，そのまま画面に表示されるプログラムで，伝言板やキーボードの入力練習などに使えます。

また，バイオリズムによるコンピュータ占いがあり，占う日のコンディションや誕生日からの日数などを計算してくれます。

つまり，ファミリーベーシックには，BASICプログラムが4つついているわけです。このプログラムは，それなりに楽しく，親しみの持てる内容ですし，改造することもできます。

また，ファミリーベーシックは，プログラムエリア（プログラムに使えるメモリの大きさ）が2Kバイトとふつうのパソコンに比べて1/8以下ですから，本格的にプログラムを作りはじめると，ちょっときゅうくつな思いをするかもしれません。



そこで，ファミリーベーシックを使いやすくし，また命令を強化したのが，ファミリーベーシックV3なのです。

V3は，ファミリーベーシックの上位バージョンとして作られていますから，ファミリーベーシックで作ったプログラムは，全部V3でも動きます。

V 3は，BASICカセット単体で発売されていますので，キーボードは今までのファミリーベーシックのものを使用します。

次に，V 3の主な特徴について説明しましょう。

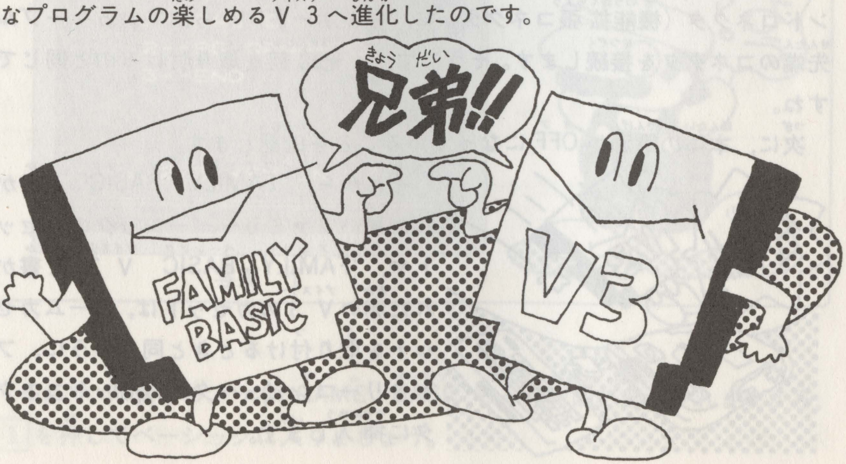
①4つのBASICプログラムを内蔵している
②BG グラフィックの使い方が変わった
③メモリ容量が2倍（4K）になった
④BASICの命令が26種類ふえた

V 3では，ファミリーベーシックのオマケの代わりに，4つのBASICプログラムを内蔵していて，ワンタッチのキー操作で実行できるようになっています。4つのうち3つまでは，本格的なゲームプログラムですから，実行して遊ぶほかに，プログラムを変更したり，作り直したりして遊ぶことができ，また，BASICプログラムの作り方を勉強するのによい材料にもなります。

また，オマケのプログラムがなくなったため，電源をONするとすぐにBASICモードとなり，今までのように質問に答えたり，名前を入力したりしなくてすむようになったため，BASICと本格的に取り組みたい人には便利です。

さらに，BASICモードとBG グラフィックモードの切換えに使用していたメニュー画面もなくなり，BASICモードからBG グラフィックモードへは，新しいBASIC命令で移行できるようになりました。BG グラフィックモードからも，キー操作によって直接BASICモードへ戻れます。

以上のような改良，強化によって，ファミリーベーシックは，より本格的なプログラムの楽しめるV 3へ進化したのです。



第2章
こうして動かすベーシック



1 キーボードとベーシックROMカセットの接続

ファミリーコンピュータで，ゲームカセットを楽しんでいるキミなら，コンピュータ本体の電源や，テレビとの接続はＯＫだと思います。ですからここでは，ファミリーベーシックを使う場合に必要な製品についてだけ説明することにしましょう。

といっても，キーボードをつなぎ，ベーシックROMカセットを入れるだけなのですが……。

キーボードは，ファミリーコンピュータ本体の前面にある小さなエキスパンドコネクタ（機能拡張コネクタ）に，キーボード側から出ているケーブル先端のコネクタを接続します。そうです！　光線銃を取り付けるのと同じですね。

次に，本体の電源がOFFになっていることを確認します。



それから，“FAMILY　BASIC”と書かれた黒いファミリーベーシックのカセットや，“FAMILY　BASIC　Ｖ３”と書かれた赤いＶ３のカセットは，ゲームカセットを取り付けるときと同じように，ファミリーコンピュータのカセットコネクタに挿入します。

2 ファミリーベーシックをスタートさせよう

接続が終わったら，ファミリーコンピュータの電源を入れてみます。そうすると，テレビ画面にコンピュータのカラーパネルが表われ，ピコピコと青が点滅します。

ファミリーベーシックでは，ベーシックのほかに，カリキュレータボードやミュージックボードなどのプログラムが遊べるようになっています。そのために，ベーシックに入れるまでに，名前を入れるなど，いろいろな操作を行わなくてはなりません。

ここでは，ベーシックのことしか説明しませんので，あまり関係のない操作は省　略します。そこで，一足飛びにベーシックに入れる操作について説明しましょう。

電源を入れて，カラーパネルがテレビ画面に表われたら，

キーボードの T キーを押しながら，本体の RESET スイッチを一度押し，その後 T キーを離します。

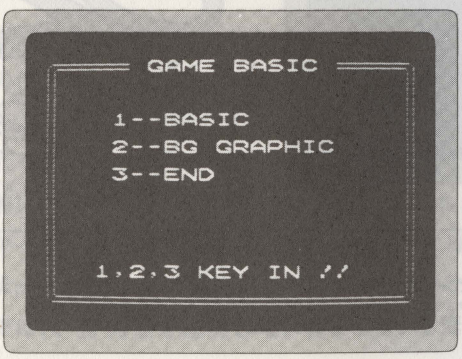


写真　1



これで，写真1のような，GAME　BASIC モード画面が表示されますから，

1 を押してベーシックに入りましょう。

V 3の場合は，電源をONにするとすぐにベーシックモードに入りますから，特別な操作はいりません。

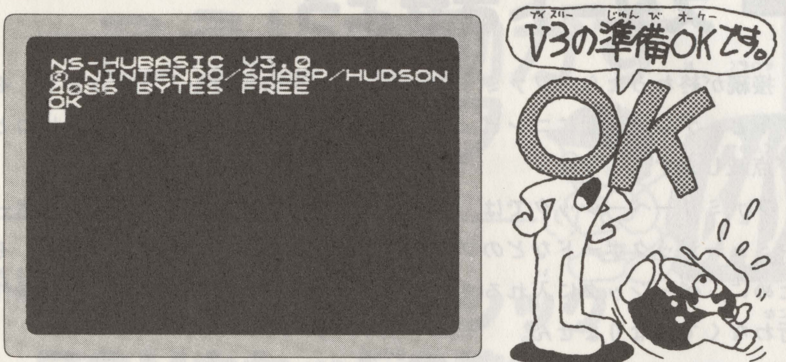


写真　2

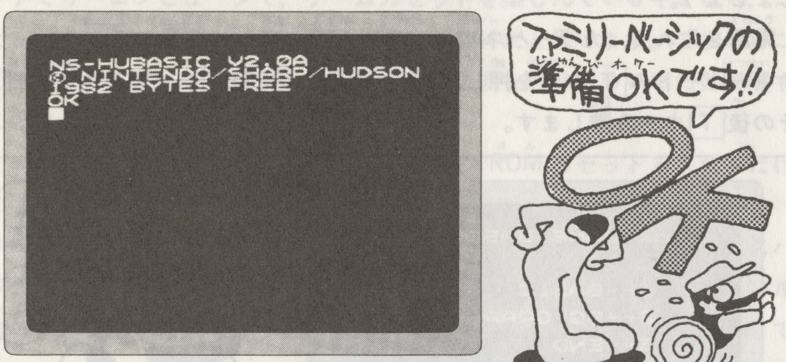


写真　3

どちらの場合も，うまくベーシックに入れると，画面の上の方にベーシックに関するメッセージが3行表示され，次の行に，

OK

■

と表示されます。“OK”の次の行に表示されて点滅している4角は，“カーソル”と呼ばれるものです。キーボードを押すと，コンピュータに文字が入り

ますが，その文字はこのカーソルのある位置に表示されます。

V3の場合，もしキーボードを接続しないで電源をONすると，

"キーボード　ヲ　セツゾクシテクダサイ"

というメッセージが表示されますので，キーボードのコネクタを接続してください。また，バックアップスイッチ（ベーシックカセットの前面にあるスライドスイッチ）がONになっていると，

"バックアップスイッチ　ヲ　OFF　ニ　シテクダサイ"

というメッセージが表示されますので，スイッチを左にスライドさせて，OFFにしてください。

3 キーの名前と押し方

ファミリーベーシックのキーボードには，いろいろな文字が書かれています。

プログラムを入力したり，ゲームで遊んだりするには，ある程度キーボードの使い方を覚えておかなければなりません。というのは，あるキーの上には4つもの文字が書かれていますし，キーの上に書かれていない文字を入力したりすることもできるからなのです。ですからプログラムのことを勉強する前に，まずキーボードの使い方を覚えておきましょう。

キーボードから文字を入力するモードには，ノーマル，ノーマルシフト，カナ，カナシフト，カナグラフィックシフトの5つがあります。

では入力方法を説明していきましょう。

① ノーマルモード

ノーマルモードというのはアルファベットの大文字が入力されるモードで，SHIFTキー，カナキー，GRPHキーが押されていない状態をいいます。

入力できる文字は，図1を参照してください。

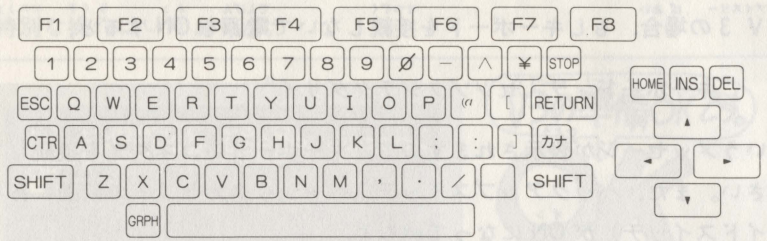


図1　ノーマルモード

プログラムを作る場合，命令語などは，ほとんどこのモードで入力できます。

② ノーマル／シフトモード

①のノーマルモードで SHIFT キーを押すと，図2に示すような文字が入力できるようになります。

SHIFT キーを押しながら，ほかの文字キーを押してください。

＝　＋　＜　＞　などの記号が入力できます。

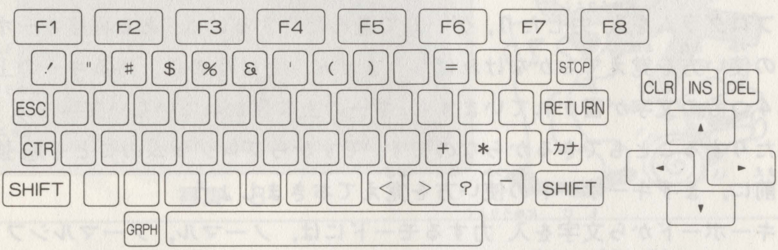


図2　ノーマルシフトモード



③　カナモード

①のノーマルモードで[カナ]キーを一度押すと，カーソルがアンダーラインにかわり，図3に示すカナ文字が入力できるようになります。カナは左上からアイウエオ順に配列されています。

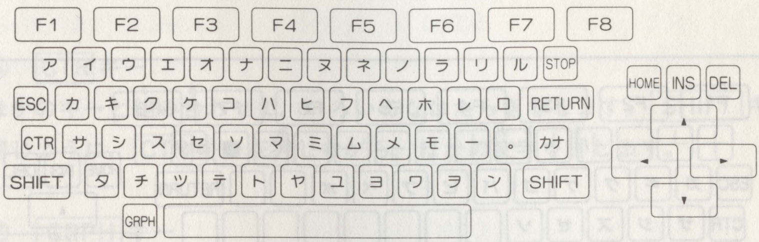


図3　カナモード

④　カナ／シフトモード

③のカナシフトモードで[SHIFT]キーを押すと，図4に示すようになります。主に残りのカナ13文字，カギカッコなどを入力する場合に使います。[SHIFT]キーを押しながらほかの文字キーを押してください。

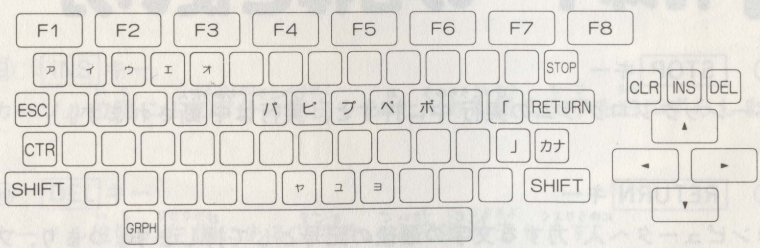
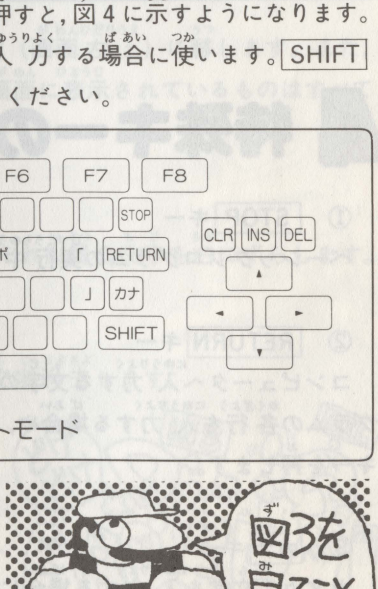


図4　カナシフトモード

カナ（シフト）モードからノーマルモードに戻るには，もう一度[カナ]キーを押してください。



⑤　カナグラフィックシフトモード

このモードは，カナロックされているときに，GRPH キーを押すことにより，図5に示すようなグラフィックキャラクタとカナ（濁音）を入力することができます。グラフィックキャラクタは，その一部だけ（罫線など）が入力できますが，キーの上に書かれていないので，わすれやすいかもしれません。

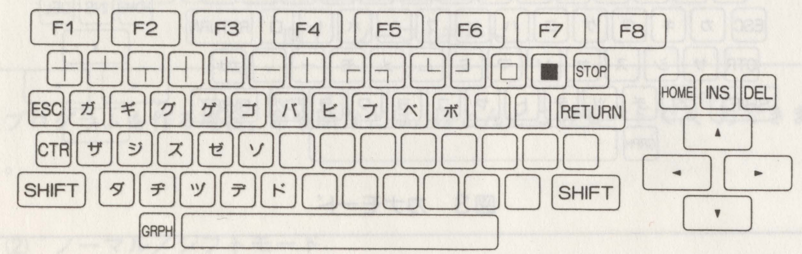


図5　カナグラフィックシフトモード

4 特殊キーの名称と使い方

①　STOP キー

ベーシックプログラムの実行中に押すと，実行は中断されます。

②　RETURN キー

コンピュータへ入力する文字の最後の記号として押します。つまり，プログラムの各行を入力する場合や，データを入力する場合などは最後にこのキーを押します。

③　カナ キー

カタカナ文字を入力する場合に押します。もう一度押すとカナ文字入力は解除されます。

④ SHIFT キー

ほかの文字キーと組み合わせて，記号やカナ小文字を入力します。

⑤ ECS キー

ベーシックの内部では使用しません。

⑥ CTR キー

ほかのキーと組み合わせて，いろいろなはたらきをします。SHIFT キーと同じように，CTR キーを押しながら，ほかの文字を押します。

⑦ GRPH キー

ほかのキーと組み合わせて，カタカナの濁音や一部のグラフィックを表示させることができます。

⑧ CLR/HOME キー

そのまま押すと，カーソルをホーム位置（画面左上）に移します。また，SHIFT キーを押しながら押すと，テレビ画面に表示されているものはすべて消え，カーソルはホーム位置に移ります。

⑨ INS キー

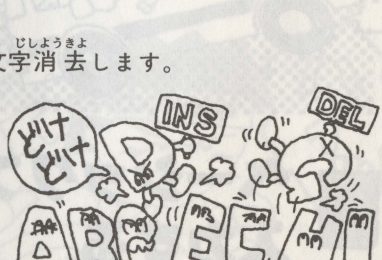
カーソル位置に1文字分の空白をあけ，それ以降の文字を右にずらします。

⑩ DEL キー

カーソル位置の左側にある文字を，1文字消去します。

⑪ カーソルキー

▲ ▼ ◀ ▶ のカーソルキーは，カーソル位置を画面上で上下左右に動かします。画面表示されている文字は，変化しません。

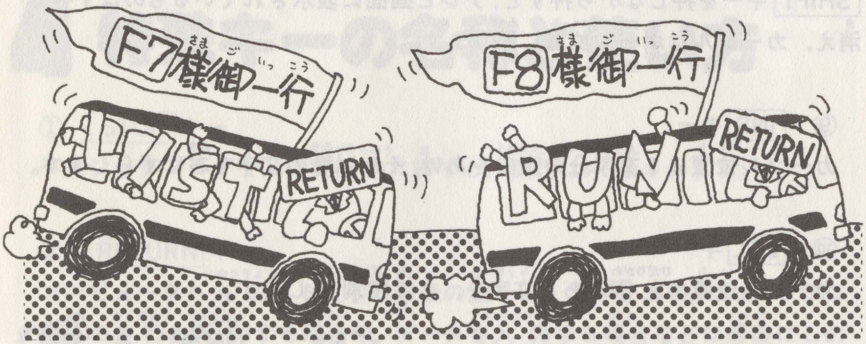


⑫　ファンクションキー（F1～F8）

あらかじめベーシック命令語が定義されているので，キーを一度押すだけで，その命令が入力，または実行されます。各キーに定義されている内容は以下のとおりです。

＜ファンクションキー＞	【ファミリーベーシック】	【v 3】
F 1	LOAD RETURN	GAME 0 RETURN
F 2	PRINT	GAME 1 RETURN
F 3	GOTO	GAME 2 RETURN
F 4	CHR$(	GAME 3 RETURN
F 5	SPRITE	SPRITE
F 6	CONT RETURN	LOAD RETURN
F 7	LIST RETURN	LIST RETURN
F 8	RUN RETURN	RUN RETURN

たとえば，ふつう P R I N T と5回押さなくてはならないところを，F 2 キーを1回押すだけでPRINTと入力できます。



5 ベーシックプログラムを入力する

ベーシックプログラムを入力するっていったって，キミはまだベーシックについても，プログラムについてもあまりよく教わっていないんだから，無

理というものだよね？

ここでは，ベーシックプログラムの内容ではなくて，“どんなふうに入力すればいいのか？”ということだけを説明しましょう。

ファミリーベーシックのはたらかせ方には2種類あります。それは，

(1)ダイレクトモード

(2)プログラムモード

この2つです。ダイレクトモードというのは，ベーシックの命令語をキーボードから入力し，その場で実行させます。命令語を入れ，RETURN キーを押したらすぐに実行されます。ふつう，“電卓的な使い方”などともいわれています。

たとえば，

P R I N T　3 ＋ 5　RETURN

と押してみてください。

```
8
OK
```

と表示されましたね。これは，PRINT 3＋5（つまり3＋5の計算結果を表示しなさい）という命令が，すぐに実行されたからです。

ところが，もう1つのプログラムモードは，正反対の性格を持っています。プログラムは，“実行する”という命令（RUN 命令）を入れてやるまでは何もしません。実行させるまでは，プログラムのリスト（一覧表：内容を順に表



示したもの）をとったり（LIST 命令），プログラムの修正を行ったりできるのです。

では，プログラムモードとダイレクトモードのちがいはどこにあるか，というと，それは行番号です。

入力された行の最初の文字が数字であると，それは行番号とみなされ，プログラムの扱いとなります。ためしに，メチャメチャな内容を入れてみましょう。

12ABC35 ＊ KJ033 [RETURN]
（今度は何も表示されません。）

そして，プログラムの内容リストを表示する命令（LIST）を入れてみましょう。

LIST [RETURN]　または [F7] を押すと，

12　ABC35 ＊ KJ033

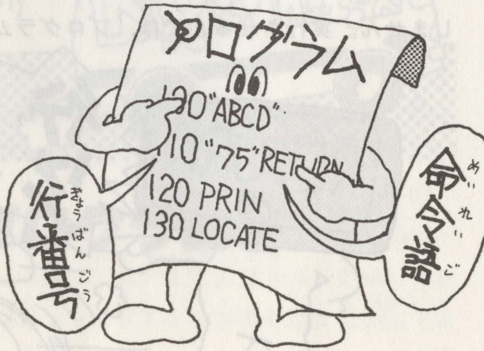
ホラ，最初の数字が切り離されていますね。これは行番号の扱いとなっているからなのです。つまり，まずは数字（行番号）を，そして命令語を入れていけばよいのです。

行番号というのは，プログラムが実行されるときの順番を示す数字です。ですから，むやみに付けるわけにはいきません。ちゃんと実行したい順序に付けなくてはなりません。

つまり，プログラムというのは，

行番号　　命令
行番号　　命令
⋮　　　　⋮
行番号　　命令

となっているもののことです。もちろん，たった1行でもプログラムです。



```
100　PRINT　3＋5　[RETURN]
```

と入れてみましょう。答えはまだ出ませんね。では，

```
RUN　[RETURN]
```

と入れてみましょう。あっ！

```
8
```

と答えが表示されました。それでは，もう一度実行してみましょう。

```
RUN　[RETURN]
8
OK
```

どうです，何度でも実行できますね。この，何度でも繰り返し実行できるというのは，プログラムの大きな特徴です。ダイレクトモードでは，同じことをさせるには，毎回同じ命令をはじめから入力し直さなければなりませんが，プログラムモードでは，1つのプログラムは，何度でも実行させることができます。

また，プログラムは，カセットテープに記録しておき，あとになって呼び戻すこともできます。ですから，長い複雑な仕事をさせるには，ダイレクトモードよりもプログラムモードの方が向いている，といえるわけです。

行番号は，プログラムを修正追加したり削除したりする場合に用いられますから，あまりメチャメチャな数字を使うのはヤメましょう。

プログラムに行番号が付くことはわかりました。では，命令の方は？

行番号は，1，5，10，でも　100，200，300でも，あとで便利なように数字が順番にならんでいればいいだけですから，かんたんです。ところが命令の方は，といえば，画面制御命令やサウンド制御命令，ムーブ命令やスプライド制御命令など，70以上の種類があり，とてもすぐには覚えきれません。

でも，よく使われる命令は決まっていますから，第3章で説明する基本命令さえマスターしてしまえば，あとはとってもかんたんに，らくらくプログラムが作れるようになります。とにかくやってみること！　それが大切です。

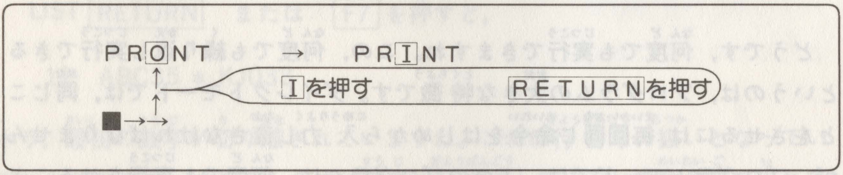
6 まちがって入力したときの訂正方法

キーボードにはもう慣れましたか？ だいぶ慣れたつもりでも，キーの押しまちがえということはよくあるものです。

そこでここでは，キーを押しまちがえたときは，どう訂正したらいいのかについて説明しましょう。

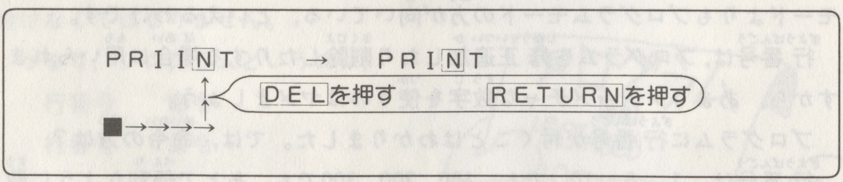
① 入力した文字の変更

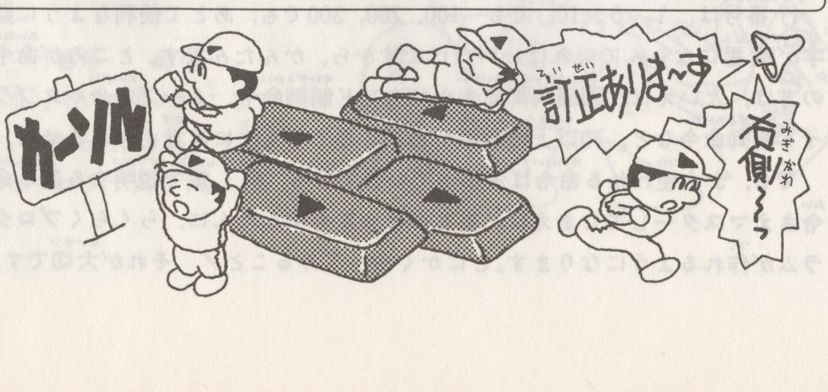
カーソルキーで，変更したい文字までカーソルを動かし，正しい文字のキーを押し直し，RETURN キーを押します。



② 1文字消したい場合

消したい文字の右隣りの文字までカーソルを移動させてから，キーボードの右上端のDEL キーを押し，RETURN キーを押します。





③ 間に文字を入れたい場合

文字を入れたい位置の右側2文字のうち，右側の文字にカーソルを移動させてから，INSキーを押します。INSキーを一度押すと1文字分の空白があきます。入れたい文字分だけ押します。その後，入れたい文字を入力しRETURNキーを押します。

```
PRNT            →    PR NT          →    PRINT
  ↑      INSを押す       Iを押す        RETURNを押す
  ↑
■→→
```

④ 行の右側の文字をすべて消したい場合

文字を入力していて，行の途中の文字から行の最後まで消したい場合は，カーソルを消したい文字の先頭に移動させ，CTRキーを押しながら，Eキーを押します。これでカーソル以降，行の終わりまでの文字は消えますので，続けて文字を入力しRETURNキーを押します。

```
PRINT "ABCDEF"   →   PRINT"ABC■
          ↑
          ↑          CTR+E
■→→→→→→→→→→→          を押す
```

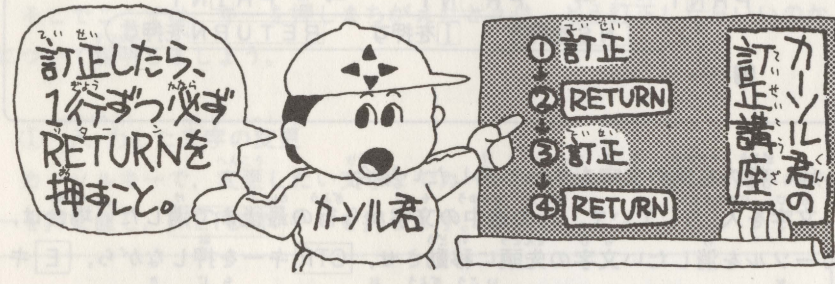
⑤ 画面の下まですべて消したい場合

文字を入力していて，行の途中から画面の下まですべて消したい場合には，CTRキーを押しながらZキーを押します。

```
100PRINT "ABCD……"        100PRINT "ABC■
110PRINT "アイウエ……"  →      画面下まで消去される
                 CTR+Z
                 を押す
```

以上は，おもに画面上での訂正方法について述べました。ベーシックプログラムを修正した場合には，修正した行で，必ず RETURN キーを押してください。RETURN キーを押さずに，カーソルキーなどでほかの場所に移動してしまうと，その修正は無効となってしまいます。

また，⑤の場合，表示は消えても，修正中の行以外はそのまま残ってしまいます。



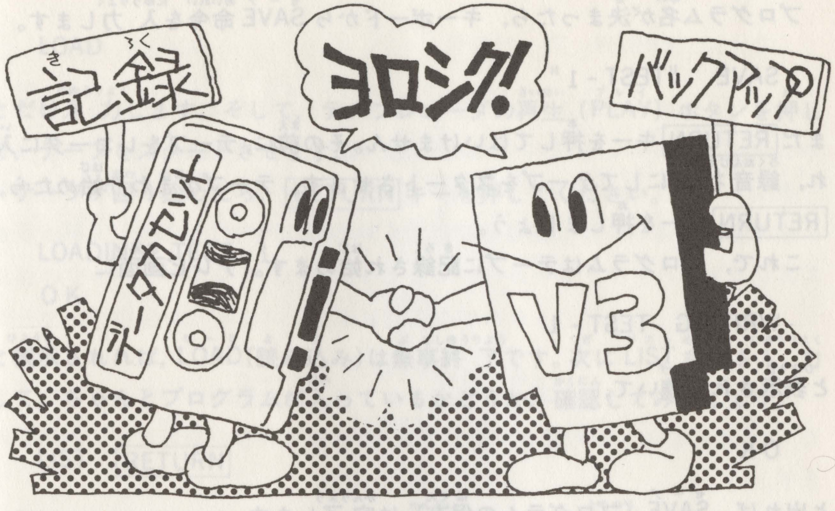
7 データカセットの使い方

キーボードからせっかく打ち込んだ長いプログラムも，電源をOFFにすると消えてなくなってしまいます。ゲームカセットのように，いつも同じプログラムを実行する場合には，最初からプログラムを記憶させ，電源を切っても消えないように作っておくことができます。ところが，ベーシックのプログラムの場合は，使う人の目的によって，いろいろなプログラムを入力し実行できるようにしておかなければなりません。このために電源をOFFにすると，プログラムの内容が消えてしまいます。

そこで，ファミリーベーシックでは，ベーシックカセット内に乾電池を内蔵させ，プログラムのバックアップ（電源を切っても，プログラムの内容を残しておくこと）ができるようになっています。

ただし，これはカリキュレータボードやミュージックボードなどの，ほかのプログラムでも使いますから，いつも安心していられるとは限らないのです。それに，ちょっと操作をミスすると，アッというまにプログラムは消えてなくなってしまいます。

というわけで，メモリバックアップをするにしても，ひとまずカセットテ



ープに記録しておけば完全です。

ファミリーベーシックのプログラムを記録するためのカセットレコーダは，できるだけ専用データレコーダ(HVC - 008)を使用してください。ラジカセやふつうのテープレコーダを使用することもできますが，読み込むときにうまくいかないものがあります。とくに，ステレオ用の高級カセットデッキなどは，かえってうまくいかない場合が多いようです。

テープレコーダを用意したら電源をつなぎ(または電池を入れ)，キーボードと接続します。ケーブルの接続方法は，次のとおりです。

キーボード側		レコーダ側
WRITE　端子	←――――→	SAVE（または MIC）端子
READ　端子	←――――→	LOAD（または EAR）端子

使用するカセットテープについても，あまり高級なものは避け，ふつうの会話用などのノーマルテープ（ノーマルバイアス）を選んでください。

プログラムをキーボードから入力したら，そのプログラムに何か名前(プログラム名)を付けます。名前は16文字以内で付けます。たとえば，「TEST-1」という名前を付けたとします。

プログラム名が決まったら，キーボードからSAVE命令を入力します。

SAVE　"TEST－1"

まだ RETURN キーを押してはいけません。その前に，テープをレコーダに入れ，録音状態にしてテープをスタートさせます。テープがまわり始めたら，RETURN キーを押しましょう。

これで，プログラムはテープに記録され始めます。テレビ画面に

WRITING　TEST－1

と表示され，続いて，

OK

と出れば，SAVE（プログラムの保存）は完了します。

プログラムの保存が終わったら，必ず正しく保存されたか確認してください。確認の方法は，テープを記録したときの先頭に巻き戻して，

LOAD?　"TEST－1" RETURN

とキーボードから入力します。正しく保存されている場合は，

OK

と表示されます。また正しく保存されていない場合は，

? TP ERROR

と表示されますので，もう一度，SAVEから繰り返えしてください。

こうしてテープに記録したプログラムを読み込むときには，LOAD命令を使います。テープを，記録したときの先頭に巻き戻してください。

次に，

LOAD　"TEST－1"

とキーボードから入力します。もし使用しているテープに，プログラムが1つしか入っていない場合や，いちばん最初のプログラムを読み込みたい場合は，プログラム名を省略して，

LOAD

とだけ入力します。そして，テープレコーダの再生（PLAY）ボタンを押して，テープをスタートさせます。

テープが回り始めたら，RETURN キーを押してください。

LOADING　TEST－1
OK

と表示されれば，LOAD（読み込み）は無事終了です。次に LIST 命令を入力して，ちゃんとプログラムが入っているかどうか，確認してみましょう。

LIST　RETURN

OK でしたか？　プログラムは，まだ完成していない途中までのプログラムでも，SAVE／LOAD することはできますから，長いプログラムを途中まで入力して「あとは明日！」というときにも，カセットテープを利用してください。

8 プログラム実行！でもエラーになったら…

「本にでているプログラムを，そのまま入力してもエラーになるのに，自分でプログラムなんか作ったら，いったいどうなるの？」なんて心配は無用です。エラーを出したくらいでは，ファミリーコンピュータはこわれません。エラーが出て，「なぜエラーが出るんだろう？」と考えるのも勉強になるのです。どんどんエラーを出しましょう！

とはいっても，プログラムが動かないのも困りものです。もし，プログラムの内容そのものにまちがいがないのにエラーが出るとしたら，それは入力のしかたが悪いのです。

そういう場合に考えられるのは，次のようなエラーです。ある行番行で入力をまちがえるとプログラム実行中に起こったエラー，

?×× ERROR IN 行番号

のように，エラーの起こった行番号を表示して止まります。ただし，表示された行番号の中に，必ずエラーの原因があるとはかぎりません。

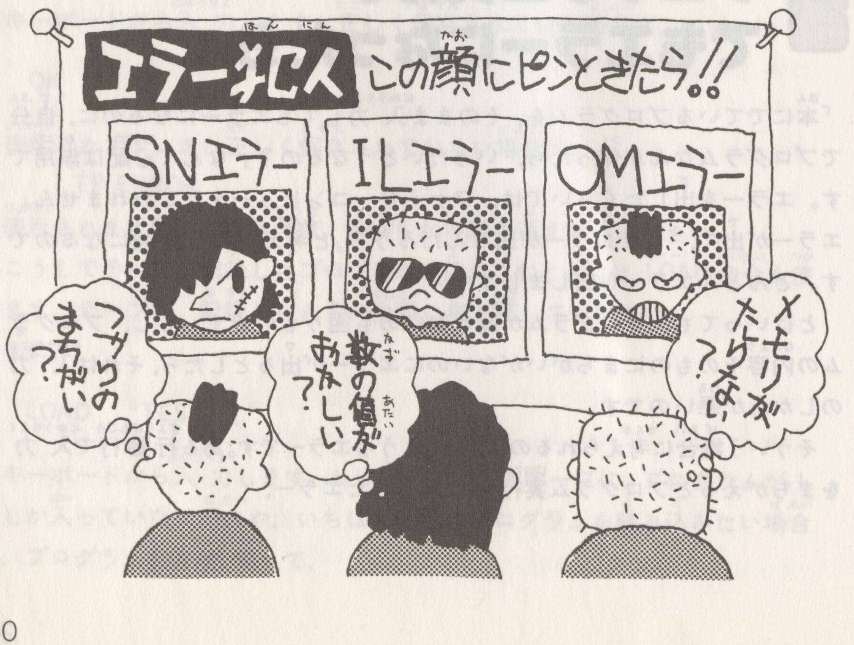
表示された行番号のプログラムがどこもまちがっていない場合は，その行に出てくる変数名と同じ変数名が出でくるそれ以前の行をさがして調べてみてください。

?SN ERROR

文法がまちがっている場合に表示されます。命令語のつづりをまちがえたり，命令語に必要な記号などが正しく入力されていない場合などに表示されます。

?IL ERROR

命令語に付随して入力する，画面上の位置やキャラクタ番号などの値が，許されている範囲を超えて指定された場合などに表示されます。



?OM ERROR

メモリが不足している場合に表示されます。ファミリーベーシックでは，メモリの容量が少ないために，よく起こるエラーです。ファミリーベーシックでは，行番号の後の空白(スペース)，命令と記号，変数などとの間の空白はつめて入力することができます。こうすると，画面上はリストが見にくくなるのですが，プログラムが必要とするメモリは少なくてすみます。このエラーが出た場合は，なるべく不要な空白をつめるようにしてください。

第3章

基本コマンドでOK!



ファミリーベーシックは，パソコンのBASICとはちがって，ゲームを作って遊ぶことを主な目的に考えて作られています。ですから，ビジネスマンのお父さんが使うのにはあまりむいていないといえます。ビジネスむけのBASICでは，200種類以上のコマンドを持っているものがありますが，ファミリーベーシックでは，70種あまり，V 3でも約100種です。

けれども，たとえばコマンドが200あったとしても，いつも全部のコマンドを使ってプログラムを作るわけではありません。1つのプログラム中で使われるコマンドの数は，50以内がふつうです。そのうちでもよく使われるのは半分くらいです。さらに，そのうちで絶対に覚えていなくてはならないコマンドは，10種類ぐらいでしょう。これ以外の命令は，忘れてしまったら説明書を読めばいいのです。とにかく，ときどきしか使わないのですから。

そういうわけで，これからプログラムを作る上で，絶対に必要な基本コマンドについて説明します。これだけ覚えておけば，あとは自由自在。自分の思うとおりのプログラムが作れるようになります。

基本コマンド

これだけは必ず覚えよう!!

①代入文
②PRINT
③INPUT
④GOTO
⑤IF
⑥FOR～NEXT
⑦GOSUB～RETURN

（秘）

1 代入文とは？

代入文とはいったい何でしょう？　ダイニューなんて，なんだか牛乳の親せきみたいですね？

代入文は，変数を使うときに絶対に必要なものです。かんたんにいうと，変数にその内容をしまうときに使われます。でもほかの命令とはちがって，ただの＝（イコール）記号ですから，命令とはいえませんが，プログラム上，とても大切なはたらきをしますから，覚えておきましょう。

代入文は，

A＝3

とか

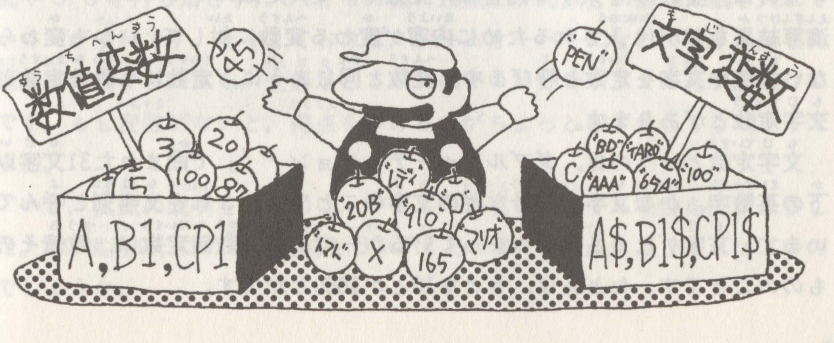
B＝A＋4

のように使います。こうすると＝（イコール）の左側（左辺）にある変数に，＝の右側（右辺）の内容が入ります。

上の例（A＝3）では，変数Aには，3という数が入ります。また，下の例（B＝A＋4）では，変数Bに，変数Aの値に4を足した数が入ります。もし，Aが3ならば，変数Bには7が入るということになりますね。

ところで，“変数”という言葉は，もう覚えたかな？　えっ，知ってるけど教えたくない？　そう，じゃあもう一回説明しておきましょう。

変数とは，1，2，3などの数値や，ABC・アイウエオなどの文字列をしまっておくための箱（記憶場所）に，英数字で名前を付けたものです。



変数は，しまっておく内容によって，次の2つにわけられます。

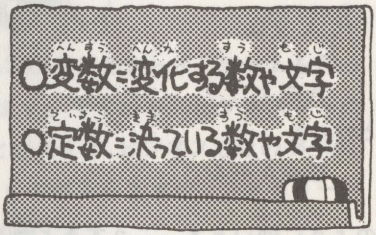
①数値変数

②文字変数

これらのちがいは，名前からもわかるように，数値変数は数値を，文字変数は文字をしまっておく場所だということです。

また，この2つの変数は，プログラムの中で区別して使わなければなりません。今までに出てきた数値変数名は，英数字で構成されていますが，文字変数名は，変数名（英数字）の後に「＄」を付けます。

例えば，



A
B1 } 数値変数名
CP1

A＄
B1＄ } 文字変数名
CP1＄

となるわけです。

数値変数に代入できるのは数値だけ，文字変数に代入できるのは文字だけです。ですから次のようにその種類をまちがえた場合はエラーとなります。

A＝"ABCD"

B＄＝10

また，変数は代入文によって，その内容を変えない限り，前に代入した内容をずっと持ち続けています。

次に変数の反対の"定数"というのもついでに説明しておきましょう。というのは，変数に代入する内容は，変数か，この定数だからです。

演算結果などが代入されるために内容が変わる変数に対して，いつも変わらない数値や文字を定数と呼びます。変数と同じように，定数にも数値定数と文字定数とがあります。

文字定数というのは，ダブルクォーテーション「"」で囲まれた31文字以下の英数字，かな文字，記号などの文字のことです。これを文字列と呼んでいます。文字がたくさんあつまっているからですね。数値定数は，数値そのもののことです。たとえば，5とか10とか1000とかです。

代入文には，次のような種類があります。

変数の型	代入する内容	内容の例	代入文の例
	数値定数	120	A＝120
数値変数	数値変数	B	A＝B
	算術演算式	B＊10＋2	A＝B＊10＋2
	文字定数	"ABC"	A$＝"ABC"
文字変数	文字変数	B$	A$＝B$
	文字列演算式	B$＋"サン"	A$＝B$＋"サン"

どうして代入文があるかというと，それは変数があるからです。では，どうして変数なんていうものがあるのでしょうか。

プログラムでは，はっきり決められないもの，そのときそのときで変化するものを扱うことがよくあります。

たとえば，ゲームの得点を考えてみましょう。もちろん最初は，得点0です。でも，敵の宇宙艇を1機やっつければ10点，UFOを打ち落とせば50点，というように次々と得点を加えなくてはなりません。

こういう場合には，ゲームの得点（スコア）と書いた箱を用意して，宇宙艇やUFOを打ち落とすたびに，その箱に10点とか50点とか書いた紙を入れていくのが便利です。そうしておけば，ゲームオーバーになったときに，その箱の紙に書かれた数を，全部足してみればいいわけです。この箱が変数なのです。もし変数がないと，得点をつけるのがちょっとむずかしくなりそうですね。

いま，変数の説明をするのに，ゲームの得点を計算する場合を例にしました。では実際に，この箱の名前をSCOREとすると，得点はどうすればよいのでしょうか。

得点が入るたびに

SCORE＝10

とか

SCORE＝50

とかするのでしょうか？　でも，そんなことをしたら，変数 SCORE には一番最後に入れた数（10とか50）しか残っていないことになります。これでは，得点の合計を計算できません。じゃあ，いったいどうしたら……

これには，実にうまい手があるのです。それは，

SCORE＝SCORE＋10

または

SCORE＝SCORE＋50

のようにするのです。ちょっと考えると，SCORE そのものと，SCORE に10を足したものが等しい（同じ）だなんておかしいですね。でも，これは，そう考えてはいけません。

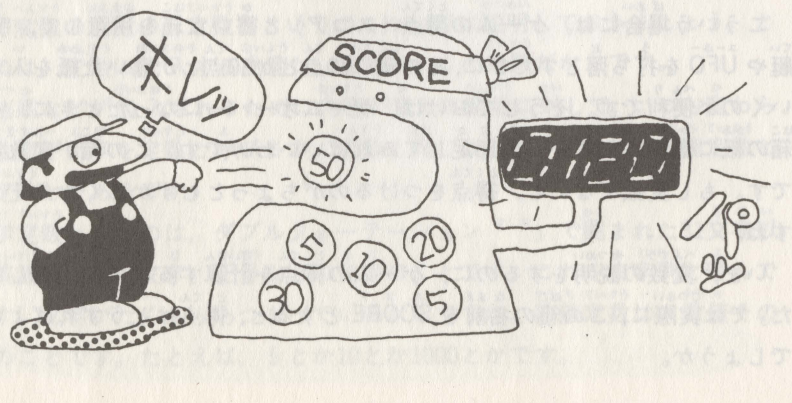
ここで使う“＝”（イコール：等号）は，ふつう学校で習う，

3＋4＝7

などという場合の“＝”とはちがいます。代入文の“＝”は，すでに説明したとおり，左辺の変数に右辺の内容を入れる，という意味ですから，

A＝A＋1

は，「変数Aの中に，それまで変数Aに入っていた内容（数）に1を足して，しまっておく」という意味になります。これでもう，得点の付け方はわかりましたね。



2 表示はおまかせ！ PRINTの使い方

プログラムを作るとき，どうしても必要になるのが，“テレビの画面に表示する”という命令です。いっしょうけんめいにコンピュータが計算しても，画面に表示されないと，何をしたのかわかりません。結果がわからなければ，何もしていなかったのと同じことですね。

PRINT 命令は，テレビ画面に，文字や数字，ときには絵も表示する命令です。

① 文字や数字（定数）を表示させる

PRINT 命令の，いちばん多い使い方は，表示させる文字や数字を直接指定して表示させる使い方です。“直接指定”というのは，前に説明した定数ということです。

では，キーボードにむかって，

```
PRINT  "ABC"
```

を入れてみましょう。そうそう，最後に [RETURN] キーを押すのでしたね。

[P][R][I][N][T] ["] [A] [B] [C] ["] [RETURN]

[T] と [C] のあとの ["] は，[SHIFT] キーを押しながら [2] のキーを押してください。

```
PRINT  "ABC"
ABC
OK
```

となりましたか？ なればケッコウ。ダメだったら最初からやり直してください。

これは，「ABC と表示しなさい」という命令に対して，コンピュータが，ちゃんと「ABC」と表示してくれたことを意味しています。

「OK」と出るのは，命令の入れ方がまちがっていなかったので，コンピュータが「OKだヨ」といっているのです。

もし，命令語の入れ方をまちがって，

```
PRONT "ABC"
```

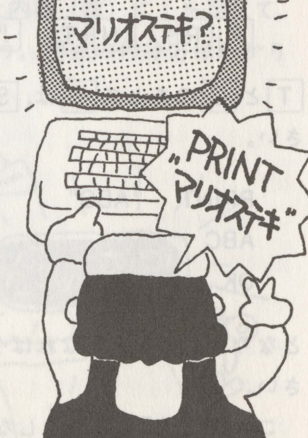
などと入れていまうと，OKとは表示されずに，

```
?SN ERROR
```

と表示されます。これは，SYNTAX ERRORの意味で，入れた命令語のつづりがまちがっていたり，文の区切りがまちがっているときに表示されるメッセージです。プログラムの場合は，IN 300のようにまちがいのあった行番号も表示してくれます。こんなのが出ても，「コンピュータにオコラレタ～～」などとしょげないで，元気を出して入れ直してください。別にコンピュータがこわれるわけではありませんし，なにしろ，キミがご主人様なのですから！

さて，めでたく「ABC」が表示されたら，" "の間（いまはABCと入っている）に，何か別な文字や数字を入れてみましょう。

```
PRINT "FAMILY BASIC"
PRINT "ファミリーコンピュータ"
PRINT "12345"
  ⋮
```



つまり，PRINT命令では，" "の間に，キーボードから入れられる文字を入れておけば，何でも表示させることができます。今は，直接命令を入力しましたが，これは，プログラムの中でPRINT命令を使った場合も同じです。

② 変数の内容を表示させる

次は，前に変数に代入しておいた内容を表示させる，という使い方です。

変数には，数値変数と文字変数があることは，前に説明しましたが，PRINT命令は，そのどちらの内容もテレビの画面に表示します。

```
A＝3
OK
PRINT  A
 3
OK
```

（＝は SHIFT キーを押しながら −ラ のキーを押してください）

プログラムにするなら，

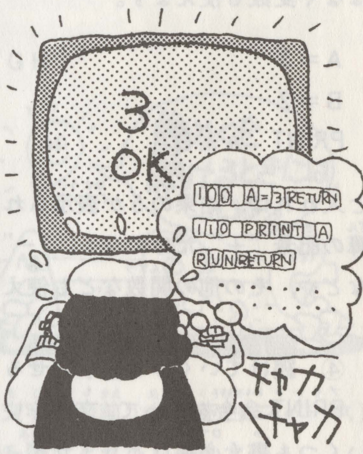
```
100  A＝3
110  PRINT  A
```

と入れておいて，RUN命令を使います。

R U N RETURN

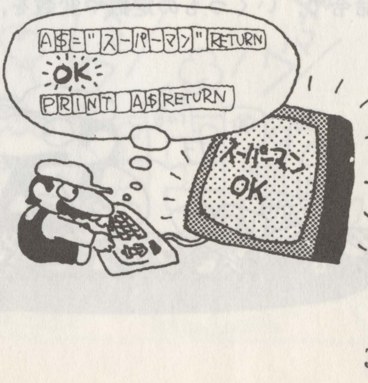
と入力すれば，

```
 3
OK
```



と表示されるはずです。文字変数も，同じようにして表示させることができます。

```
A$＝"スーパーマン"
OK
PRINT  A$
スーパーマン
OK
```



③　計算結果を表示させる

ちょっと変わったPRINT命令の使い方に，計算式を書いておいて，その結果を表示させる，というのがあります。

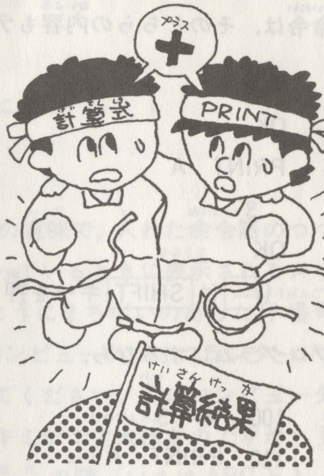
たとえば，

PRINT　3＋4

と入れると，

7

と結果が表示されます。これは，定数だけではなく変数も使えます。

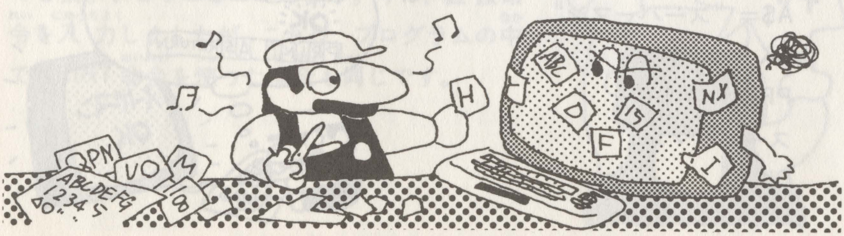


A＝3
B＝2
PRINT　A＋B

と入れれば，結果の5が表示されたと思います。もちろん，計算式には足し算の記号“＋”だけでなく，“－”（引き算）“/”（割り算），“＊”（掛け算）などや，その他の関数なども使えます。

④　続けていくつも表示させる

PRINT命令を使って，文字をいくつも表示させたい場合は，PRINT命令をいくつも書く必要はありません。こういうときは，“，”（コンマ　Mの右どなりのキー）や“；”（セミコロン　Lの右どなりのキー）を使うと，1つのPRINT命令で，いくつもの定数や変数を，ならべて書くことができます。



コンマ（,）で区切る場合は，

PRINT　"ABC"，　"DEF"，　"GHI"

のように書きます。そうすると，テレビ画面では，まず“ABC”と表示され，その同じ行の右側，行の先頭から8文字目に，“DEF”と表示されます。つまり，“,”を使うと，てきとうに文字の間隔があいて表示される，というわけです。

|←8文字→|←8文字→|←8文字→|←（残り）4文字→|
ABC　　　DEF　　　GHI

また，セミコロン（；）で区切ると，文字が続けて表示されます。

たとえば，

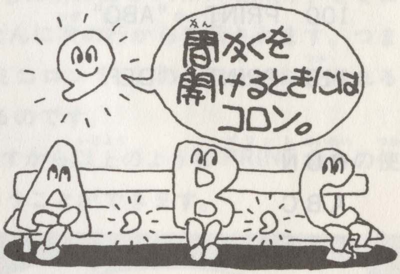
PRINT　"ABC"；"DEF"；"GHI"

とすると，

ABCDEFGHI

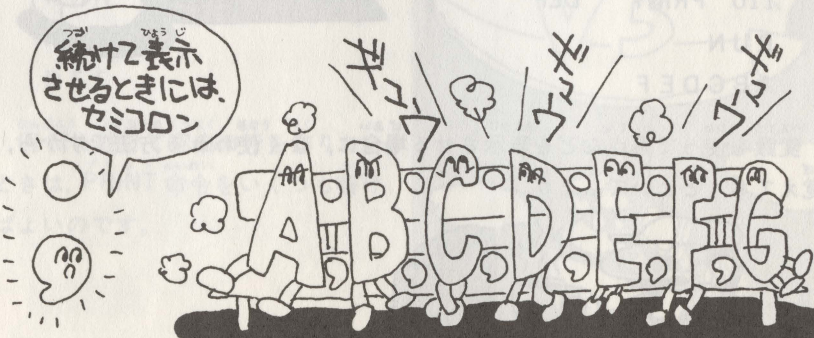
と，まるで，

PRINT　"ABCDEFGHI"



を入れたのと同じように表示されます。

もし，文字の間を少しあけたければ，各文字列の前か後に空白（スペース）を入れておくか，空（スペース）の文字列を間に入れておきます。



```
PRINT "ABC" ; " DEF" ; " GHI"
```

または,

```
PRINT "ABC" ; "␣" ; "DEF" ; "␣" ; "GHI"
```

変数を使って表示させる場合は，後の方法が便利です。たとえば，

```
PRINT A$ ; "␣" ; B$
```

のように使います。

また，セミコロン（；）をPRINT命令の最後に入れておくと，文字を表示した後，次の表示位置が次の行の画面左はじに戻らなくなりますので，別のPRINT文などで，続けて文字を表示できます。

```
100 PRINT "ABC"
110 PRINT "DEF"
RUN
ABC
DEF
```




```
100 PRINT "ABC";
110 PRINT "DEF"
RUN
ABCDEF
```

変数を使って得点などを表示させる場合に，よく使われる方法ですから，覚えておくとよいでしょう。

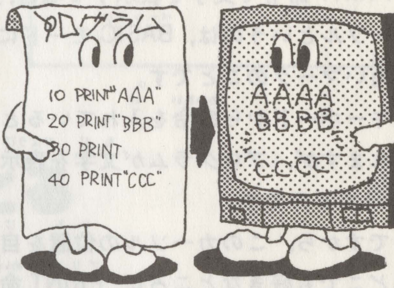


⑤ 改行する

PRINT 命令でテレビ画面にいろいろ表示させると，1 行あけたくなることがあります。そういうときには，PRINT 命令に，文字や変数を付けないで使います。そうすると，1 行あいて，次の行から文字が表示されます。

```
100 PRINT  "ABC"
110 PRINT
120 PRINT  "DEF"
RUN
ABC

DEF
```



ただし，この改行のための PRINT 命令の前にある PRINT 命令が，セミコロン（；）で終わっている場合には，たんに次の行から表示されます。つまり，こういう場合，PRINT 命令は，セミコロン（；）のとり止めに使われるため，1 行あかせることはできなくなるのです。

行を変えることを改行といいます。ですから以上のような PRINT 命令の使い方を，改行のための PRINT 命令ということができます。

```
100 PRINT  "ABC" ;
110 PRINT
120 PRINT  "DEF"
RUN
ABC
DEF
```



何行も続けて空白の行を作りたいときは，PRINT 命令をいくつも書けばよいのです。

3 PRINTの応用

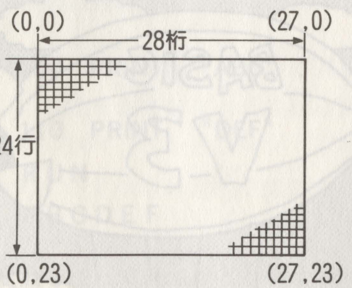
PRINT命令で文字を表示する場合，文字はカーソルの位置に表示されます。カーソルというのは，BASICモードに入ったときに，チカチカ点滅している四角いマークのことです。

キーボードから命令を入れているときも，このカーソル位置に文字が表示されますが，プログラムが文字を表示するときにも，このカーソル位置から表示するのです。

ですから，このカーソルの位置を自由に動かす命令があれば，テレビ画面のどこでも好きなところに，PRINT命令を使って文字を表示させることができるわけです。このための命令がLOCATE命令です。

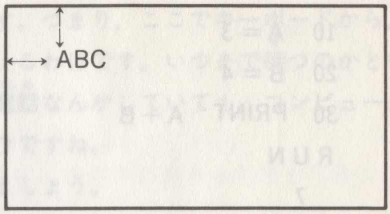
LOCATE命令は，画面上の位置をヨコ（X方向）とタテ（Y方向）に分けて考えます。ヨコは0～27までの28個，タテは0から23までの24個あります。ですから，画面の位置は28×24＝672あるのです。

タテ，ヨコの数字とテレビ画面上の位置の関係は，図1のとおりです。





たとえば，上から５行目，左から５文字目から文字（“ＡＢＣ”）を書きたければ，




```
100　LOCATE　4，4
110　PRINT "ＡＢＣ"

RUN
```

とします。

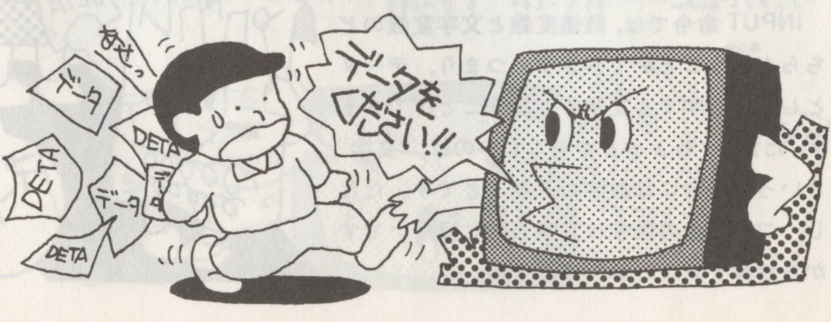
4 データを受けとる INPUT

プログラムを作るということは，コンピュータに何か（ゲームとか計算とか）をさせるということです。でも，何かをさせるためには，そのための材料が必要です。コンピュータでは，この“何か”に当たるものを「データ」と呼んでいます。

たとえば，足し算をさせるとします。

```
10　PRINT　3＋4
RUN
 7
OK
```

この場合“3”も“4”もデータですが，これはもうプログラムの中に入っていて，プログラムの実行時に変更することはできません。



では，次のプログラムはどうでしょうか？




```
10　A＝3
20　B＝4
30　PRINT　A＋B
RUN
  7
OK
```

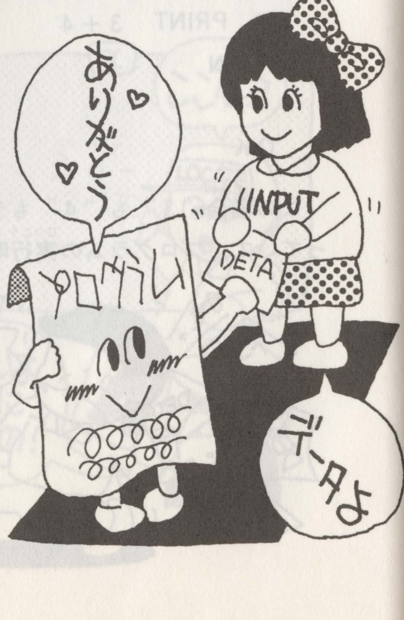
これもあまり大差ありませんね。ここで，変数AやBの値をそのつど変えることができれば，同じプログラムでありながら，いろいろな足し算ができることになります。つまり“何を”にあたるデータを変えれば，できあがりはいろいろ変えられるわけです。

このために，「データをキーボードから入れて，変数にしまう」という命令があります。それがINPUT命令です。

① キーボードからデータを読み込む

INPUT命令でキーボードからデータを読み込むには，

```
INPUT　A
または
INPUT　A$
```



のように，INPUT命令に続いて変数を書きます。そうすると，キーボードから入れられたデータは，指定した変数に代入されます。

INPUT命令では，数値変数と文字変数のどちらも扱うことができます。つまり，データとして，数字でも文字でも扱うことができます。ただし，ちょっとややこしいのは，“文字”といった場合には数字も入ることです。ただし，この場合の数字は文字と同じ扱かいですから計算することはできません。

INPUT命令が実行されると，まずそのときのカーソル位置に“？”マークが表示され，次にカーソルが点滅します。つまり，ここでキーボードから，文字や数字を入れてくれるのを待っているわけです。いつまで待つのかというと……ずっとです。キミが友達と長電話なんかしていても，コンピュータはずっと待っているのです。カワイソウですね。

では，キーボードから何か入れてみましょう。

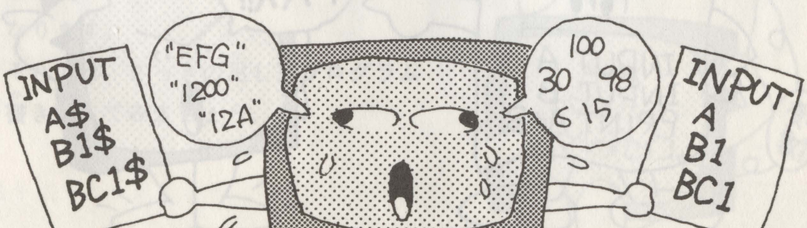
```
10  INPUT  A
20  PRINT  A
RUN
?■
```

たとえば，12とか345とか入れると(RETURNキーを忘れないように)，ちゃんとその数が表示されますね。では，もしまちがって文字を入れてしまったらどうなるのでしょう？　ABCでもXYでもいいから，もう一度RUNして入れてみましょう。答は何と出ました？　そう0ですね！　エラーになりませんが，まちがって文字を入れてしまうと，文字は0と判断されてしまうのです。

では，こんどは変数を文字変数に直して，やってみましょう。

```
10  INPUT  A$
20  PRINT  A$
RUN
?■
```

こんどは何を入れても平気ですね。いろいろな文字や数字を入れてみましょう。

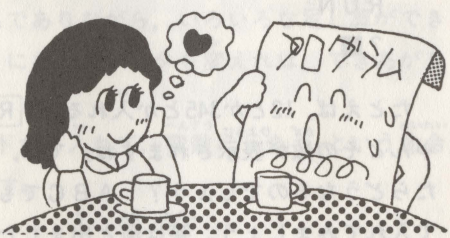


さあ，それではそろそろ，さっきの足し算プログラムができそうです。

```
10 INPUT A
20 INPUT B
30 PRINT A+B
```

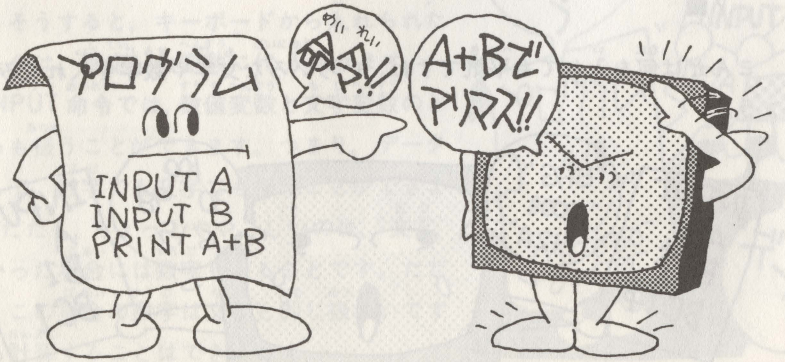
このプログラムを実行すると，コンピュータはまず変数Aに入れる数をきいてきますから，それを入れます。そうすると次に，変数Bに入れる数をきいてくるので，それを入れてください。そうすると，最初に入れた数と次の数を足して，その結果をテレビ画面に表示します。

```
RUN
? 13
? 6
 19
OK
```



どうです？ 答は合っているでしょうか？ いくらちっちゃくても，ファミリーコンピュータはリッパなコンピュータですから，まさか計算をまちがえるようなことはないと思いますよ！ いろいろためして，自分の答とくらべてみてください。

そうそう，こんなふうにINPUT命令を重ねて，いくつもデータを入力する場合には，変数名をカンマ（,）で区切って，1つのINPUT命令で書くことができます。



ですから，さっきのプログラムは，

```
10 INPUT  A, B
20 PRINT  A+B
```

のように書くことができます。1行ぶん得しちゃいましたね。

② 画面表示後にデータを読み込む

足し算プログラムはうまくいきましたか？　たぶんうまく動いていると思いますが，でもプログラムを作ったキミは，内容がわかっているからよいとしても，キミの友達などに使ってもらうとしたらどうでしょう。

いきなり“？”マークが出てきたって，何をしたらいいのかわからないんじゃないかな？

そこで，INPUT命令では，まず最初に何かメッセージを画面表示してから，データを読み込むようにできています。表示させるメッセージは，変数名の前に"（クォーテーション）でくくり，その次に“；”（セミコロン）で区切って変数名を書きます。

そうすると，まずメッセージが画面上に表示され，そのメッセージのすぐ後に？マークとカーソルが表示され，データの入力待ちとなります。

たとえば，

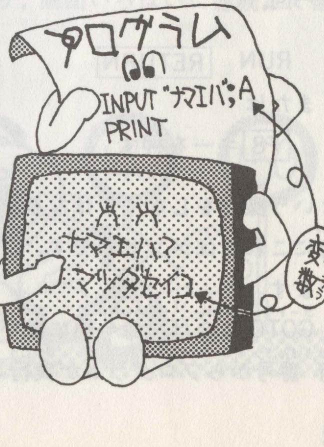
```
INPUT "ナマエ ヲ イレテ クダサイ";N$
```

とすると，

```
ナマエ ヲ イレテ クダサイ?■
```

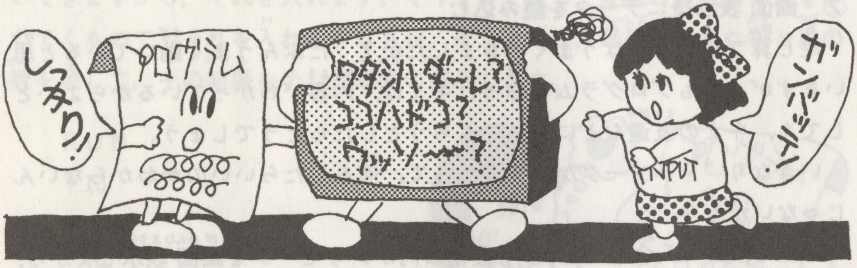
となります。

それでは，さっきの足し算プログラムを書き直してみましょう。



```
10 INPUT  "1バンメノ  カズヲ  イレテクダサイ";A
20 INPUT  "2バンメノ  カズヲ  イレテクダサイ";B
30 PRINT  "コタエハ";A+B;"デス"
```

どうです。だいぶプログラムらしくなりました。これなら友達にもわかってもらえそうですね。



5 GOTOは鉄砲玉

プログラムというのは，目的を実行するための命令を順序よく書いたものだ，というお話を前にしましたが，プログラムによっては，繰り返し繰り返し同じことをしたい場合だってあります。

たとえば，INPUT命令のところで作った，足し算プログラムは，答が出たところで終わってしまいます。もし，もう一度別な計算をさせたければ，もう一回，

RUN [RETURN]

または

[F8] キーを押す

という操作をしなくてはなりませんでした。何度も繰り返して計算したい場合にこれではメンドウです。こういうときのためにGOTO命令が用意されています。

GOTO命令は，命令語に続いて行番号を書きます。そうすると，次はその行番号からプログラムが実行されます。

GOTO命令で，いわば行番号をジャンプさせるわけですが，その行き先はGOTO命令のある行自身でもかまいません。

ただし，そうしておくとプログラムは，グルグル同じ所をまわっているだけで，いっこうに終わらなくなってしまいます。これを無限ループといいます。GOTO命令を使うときに注意しなくてはならないのは，この無限ループにならないようにすることです。

つまり，GOTO命令のある行より前の行に戻るときには，必ず繰り返しとなりますから，それでよい場合は別にして，その繰り返しから抜け出る条件を作っておかなければなりません。その条件の作り方については，次のＩＦ命令で説明します。

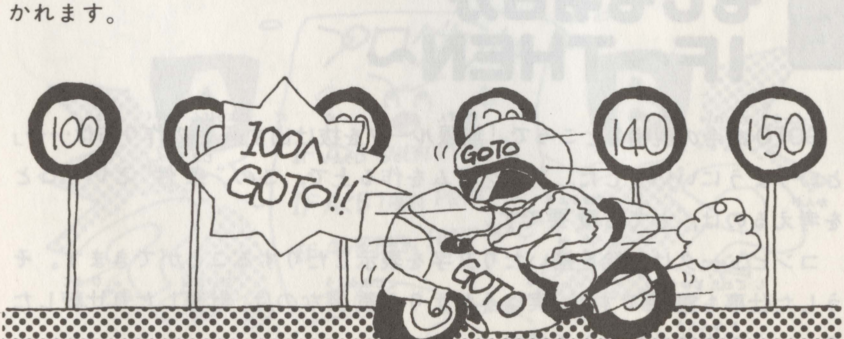
GOTO命令のはたらきがよくわかる例をあげましょう。

```
10  PRINT  "A";
20  GOTO  10
```

このプログラムを実行すると，画面はＡという文字でいっぱいになります。また，このプログラムを止めたいときには，STOP キーを押してください。こんなふうにプログラムを直してみてもおもしろいかもしれません。

```
 5  INPUT  "アナタノ  ナマエハ";N$
10  PRINT  N$
20  GOTO  10
```

こうすると，キーボードから入力した名前が，画面いっぱいに連続して書かれます。



それでは次に，INPUT 命令のところで作った，足し算プログラムの最後の行に，

```
40 GOTO 10
```

と入れ，プログラムを実行してみましょう。

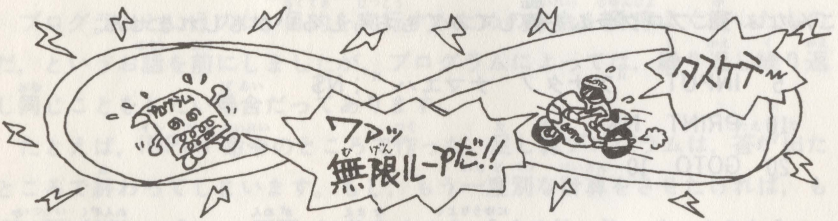
こんどはどうですか？　答が出たあともプログラムは終わらずに，また初めに戻って数字の入力をさいそくしてきました。これで，毎回 RUN する手間がはぶけたわけです。

でもチョット気になるのは，前の計算の結果と，次の計算の初めの行がくっついて見にくいことです。ここで 1 行あけておけば……。そうです。PRINT 命令のところで説明したのを，思いだしてくれましたか？　PRINT 命令を 1 つ入れておけばいいのでしたね。

それでは，GOTO 命令でプログラムの最初のところに戻る前に，

```
35 PRINT
```

と入れておきましょう。これで完成！



6 もしも明日が IF～THEN～

GOTO 命令の説明のところで，「無限ループを抜け出す条件の作り方は……」というようにいいました。プログラムを作る上で，この“条件”ということを考えるのは，とても重要です。

コンピュータは，絵を描いたり文字を表示したりすることができます。そうした仕事も重要ですが，実はそれよりも重要なのは，計算したり比較したりという基本的なはたらきです。

“計算する”というのは，コンピュータのはたらきとして，誰もが知っていることですが，コンピュータが“比較する”というのは，あまり知られていないようです。

ところが，これは本当によく行れることなのです。

“比較”というのは，たとえば次のようなものをいいます。

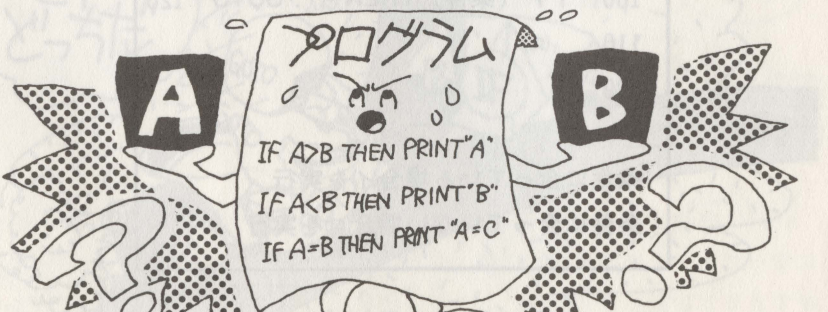
①AとBが等しい	A＝B
②AとBが等しくない	A＜＞B
③AがBより大きい	A＞B
④AがBより小さい	A＜B
⑤AがBより大きいか，または同じ	A＞＝B
⑥AがBより小さいか，または同じ	A＜＝B

これがプログラムになると，「もしAとBが同じだったら」あるいは，「もしAがBより大きかったら」などの条件となるのです。このための命令が，IF～THEN～命令です（以下IF命令と呼びます）。

IF命令は，

IF〈条件〉　THEN〈命令〉

のように使います。IF命令では，〈条件〉が整っているとき，あるいは正しかったときに，はじめてTHEN以下の命令を実行します。それ以外は，THEN以下は無視され，次の行の命令を実行します。もっとも，THEN以下にGOTO命令がなければ，THEN以下を実行したとしても，その後には規則どおりに次の行が実行されるわけです。



ですから，条件に合っていたら，(イ)の命令，ちがっていたら(ロ)の命令というように分けたい場合は，THEN以下にGOTO文を入れ，次の行をジャンプさせる必要があります。

次に，条件文の例をいくつかあげておきましょう。

```
IF  A＝3  THEN ～
IF  A$＝ "テスト" THEN～
IF  B＞5  THEN～
```

また，THEN以下にはいろんな命令が書けますが，この行だけでは書ききれないような場合には，一度GOTO命令でほかの行番号にジャンプし，いろいろ命令を実行した後，戻ってくる（次の行またはその次の次の行）ということもできます。

その場合には，

```
100  IF (条件) THEN (イ)
110  (ロ)
120    ↓
```

条件に合っている場合(イ)(ロ)と実行
条件に合っていない場合(ロ)を実行

```
100  IF (条件) THEN(イ)：GOTO 120
110  (ロ)
120    ↓
```

条件に合っている場合(イ)を実行
条件に合っていない場合(ロ)を実行

IF～ THEN GOTO 行番号

となるわけですが，このうちの“THEN”または“GTTO”の片方は省略することができます。つまり，

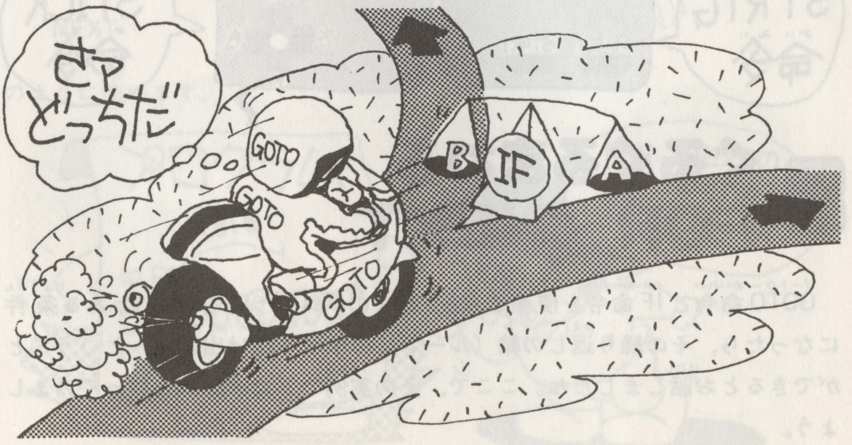
IF～ THEN 行番号
IF～ GOTO 行番号

とすることができます。

次のプログラムを入力してみましょう。

```
100  INPUT  "1―9  ノカズヲイレテクダサイ"; A
110  IF  A>5  THEN  S$="オオキイ" : GOTO  130
120  S$="チイサイ"
130  PRINT  S$
140  PRINT
150  GOTO  100
```

これは，1から9までの数を，キーボードから入力してもらい，その数が5より大きければ（6以上なら）"オオキイ"と表示し，5以下なら"チイサイ"と表示するプログラムです。110行の変数Aの条件を，いろいろ変えてためしてみましょう。



7 IFの応用

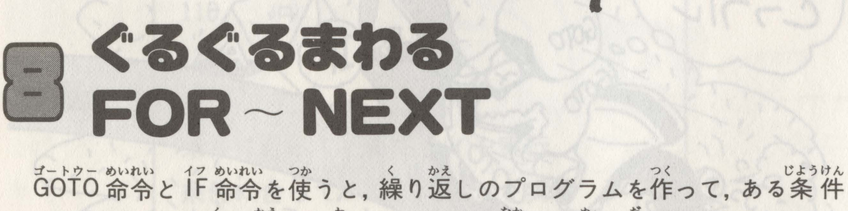
条件によって，次の命令を変える，ということは理解してもらえたと思います。でも，「条件といっても数値や文字列の比較ばかりじゃ何に使っていいのかわからない」という人もいるかもしれませんね。そういう人は，きっと，“コントローラのSTARTボタンが押されたら”とか，“コントローラの✚ボタンの上の方が押されたら”というのがやりたいのじゃないのかな？

もちろんそれは大丈夫！　そのためにSTRIGやSTICKという命令があります。

STRIGを使うと，コントローラのSELECTやSTART，Aボタン，Bボタンを押したときには，その押したボタンに対応した値が得られます。たとえば，Aボタンが押されていれば，8というようにです。

また，STICKを使うと，コントローラの✚ボタンの上下左右の動きに対応した値が得られます。たとえば，右なら1，上は8というようにです。

これでOKですね！　数値になればしめたもの，あとはIF命令を使って，ボタンが押されたときのプログラムを考えればよいのです。



8 ぐるぐるまわる FOR～NEXT

GOTO命令とIF命令を使うと，繰り返しのプログラムを作って，ある条件になったら，その繰り返しの輪（ループ）の中から抜け出す――ということができるとお話ししましたね。ここで，その実例のプログラムを作ってみましょう。

```
100  A＝10
110  PRINT  A；" "；
120  IF  A＝20  THEN  150
130  A＝A＋1
140  GOTO  110
150  PRINT
RUN
10  11  12  ……19  20
```



このプログラムは，10から20までの数を表示するものです。Aの値と10から1ずつ足しながら表示していき，20になったらもう表示するのを止める，というプログラムです。

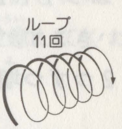
つまり，ここでは行140から行110に戻るループから行120のIF命令で抜け出しているわけです。

でもこれは，あまりカッコよくありません。なぜなら，ループに入るときの条件も，出るときの条件もはじめから決まっているのですから，もっとすっきり書けるはずです。それには，FOR～NEXT命令がピッタリです。

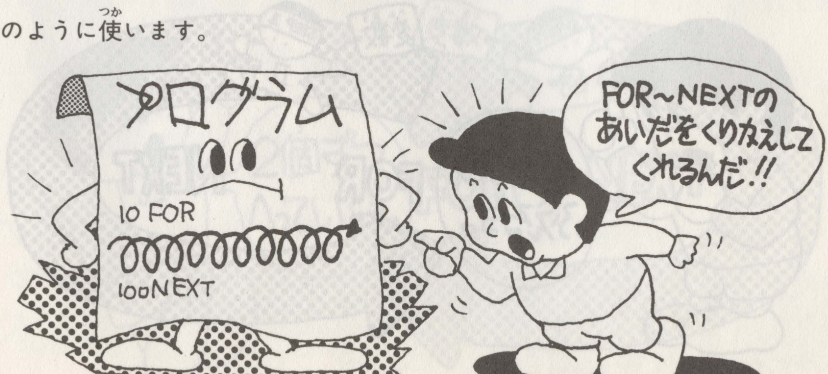
FOR～NEXT命令は，FORとNEXTを一対にして，

```
        ループ始めの値
            ↓
FOR    A＝10   TO   20←ループ終わりの値

NEXT
```



のように使います。



FOR命令は，必ず1つの変数(ループ変数と呼びます)を必要としますが，この変数は，プログラムのほかの部分で使うことは，なるべく避けたほうがよいでしょう。

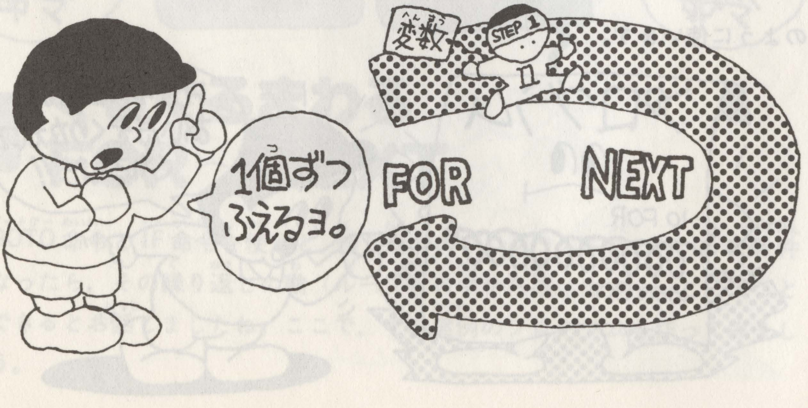
この変数の始めの値と終わりの値を決め，実行するとNEXTに出合うたびに変数の値に1が足されます。そして，変数の値が終わりの値に一致したら，ループは終わりとなり，NEXTの次の命令を実行します。

先の例をFOR〜NEXTを使って書き直すと，次のようになります。

```
100  FOR  A=10  TO  20      ┐
110  PRINT  A;" ";          │ループ
120  NEXT                   ┘
130  PRINT
RUN
10  11  12……19  20
```

どうです。だいぶスッキリしましたね。でも，変数が1つずつ増えていく場合は，これでよいとして，逆に1つずつ減っていったり，2つずつ増えたりする場合は，どうすればよいのでしょう。

そういうときには，FOR〜TOに続けて，さらにSTEPという条件を加えます。STEPは，ふつう＋1になっているため，何も指定しないと1つずつ変数の値が増えていきますが，STEPの値を−1にしたり＋2にすると，1つずつ減らしたり，2つずつ増やしたりすることができます。



先の例で，1つおきに数を表示させるには，

```
100  FOR  A=10  TO  20  STEP  2
110  PRINT  A; " ";
120  NEXT
130  PRINT
RUN
10  12  14  16  18  20
```

のようにします。また逆に1つずつ減らすには，100行を

```
100  FOR  A=20  TO  10  STEP  -1
```

とします。変数の始めの値と終わりの値も入れ替えておかなくてはならないということも覚えておいてください。

このようにプログラムを変更すると，実行結果は，

```
RUN
20  19  18  17……12  11  10
```

のようになります。

また，ほかのパソコンBASICのように，NEXTのあとにループ変数名をつける必要はなく，つけるとエラーになりますので注意してください。

```
120  NEXT  A  ←—×まちがい
```



9 お使いに行ったら戻ってきてね GOSUB～RETURN

プログラムを作っていると，何回も同じ仕事をさせる部分が必要になることがあります。これを，そのままプログラムにしてしまうと，プログラムも長くなってしまいますし，そうするとコンピュータのメモリ（記憶場所）をたくさん使ってしまいます。特に，ファミリーコンピュータのように，メモリに制約のある機種では，プログラムを短かく作ることは大切です。

このためには，すでに説明したGOTO命令を使うこともできそうです。つまり，何度も必要になる部分を，プログラムの流れ（メインプログラム）から切り離し，別の行番号を付けてもう1つの流れ（サブルーチン）を作るのです。そうして，その仕事が必要になるたびに，そのサブルーチンにGOTO命令でジャンプすればいいのです。

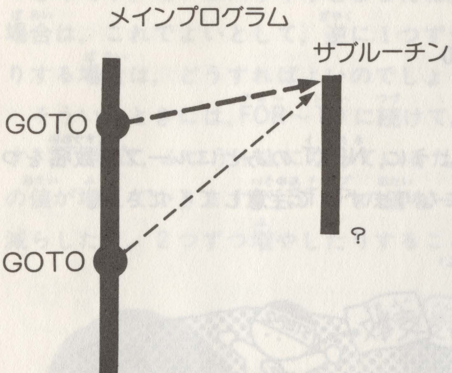


図 GR-1



ところがここで，困ったことが1つあります。サブルーチンにジャンプするのはいいのですが，サブルーチンからメインルーチンに戻るときはどうしたらいいのでしょう。えっ，それもやっぱりGOTO命令を使うのですか？　ところがそうはいかないのです。

というのは，サブルーチンは，メインプログラムの何ヵ所からもジャンプして実行されます。そうすると，サブルーチンが終わってメインプログラムに戻るときに，毎回その戻る先がちがっていることになります。ところがGOTO命令の飛び先行番号には，変数が使えませんから，これはたいへんです。本当は，サブルーチンにジャンプしたメインプログラム上のGOTO命令の，次の命令に戻りたいのです。

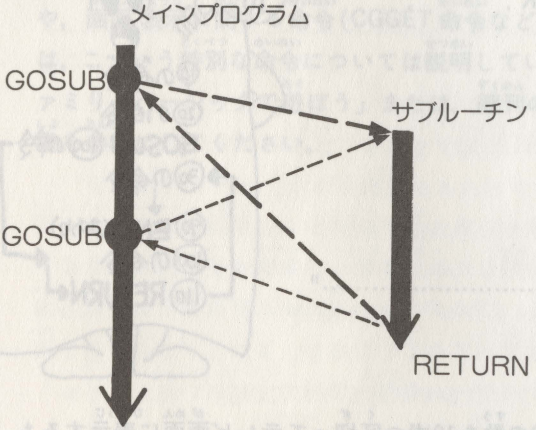
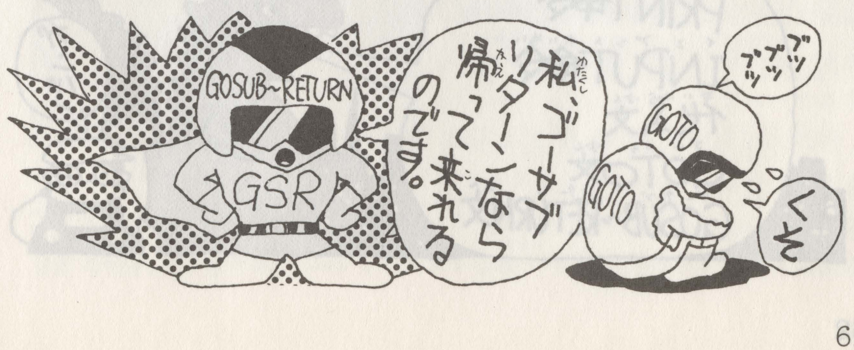


図 GR-2



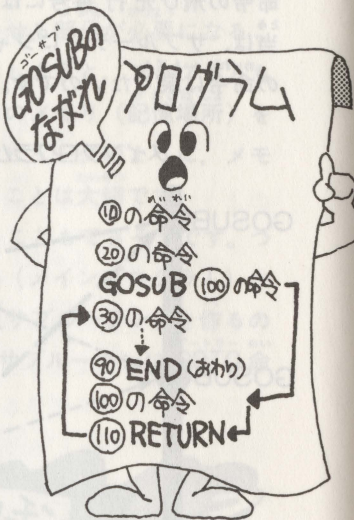
そこで，こうした用途のために考え出されたのが，GOSUB〜RETURN命令です。GOSUB命令は，GOTO命令のかわりに，メインプログラムのほうに入れておき，RETURN命令はサブルーチンのいちばん最後のところに入れておきます。

こうしておくと，メインプログラム内のGOSUB命令でサブルーチンにジャ



ンプし，サブルーチンを実行したあと，RETURN命令でメインプログラムに戻ります。

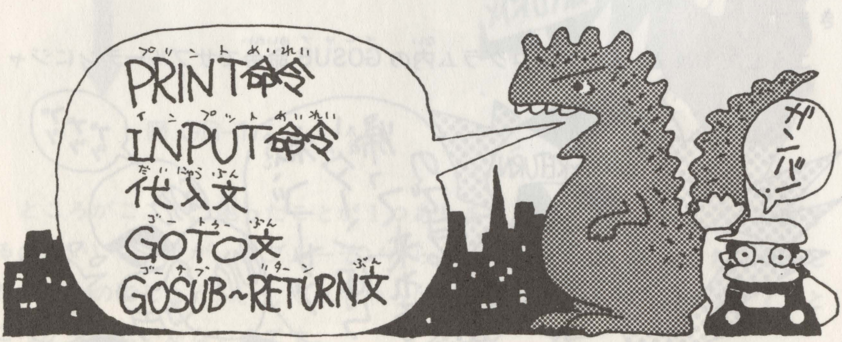
つまり，RETURN命令は，特に行番号を指定しない場合には，直前に実行されたGOSUB命令の次の命令に戻ります。




```
100  FOR  J=0  TO  9
110  FOR  K=0  TO  9
120  PRINT  J＊10+K；" "；
130  NEXT
140  GOSUB  1000
150  NEXT
160  GOSUB  1000
170  END
1000 PRINT
1010 PRINT  "......................................."
1020 PRINT
1030 RETURN
```

このプログラムは，0～99の数を10ずつ区切ってテレビ画面に表示するものです。区切りの表示（"………"）部分がサブルーチンになっていて，FOR～NEXTのループの中で，99まで表示し終わった後に使われています。

このように，GOSUB～RETURN命令をうまく使うと，プログラムを短く，スッキリスマートにさせることができます。



●

以上説明した基本命令は，代入文，PRINT命令，INPUT命令以外は，4つとも何かをさせる命令というのではなく，プログラムの実行上の流れなどに関する命令でした。ですから，PRINT命令とINPUT命令でできること以外をさせたい場合には，また特別な命令が必要になります。たとえば，マリオを歩かせたい場合には，ゲームキャラクタに関する命令（MOVE命令など）や，画面表示に関する命令(CGGET命令など)が必要となります。この本では，こういう特別な命令については説明していませんので，弊社発行の，「ファミリーベーシックで遊ぼう」または，続刊のファミリーベーシックの解説書を参考にしてください。

第4章 ゲームライブラリー

この章では，16パズルや，迷路脱出ゲームなど5本のプログラムを紹介します。

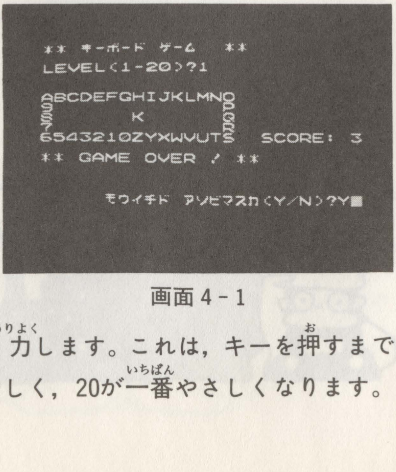
ここでは，ゲームの遊び方はもちろん，プログラムの説明，改造方法を紹介していきます。今までに，説明した以外の新しい命令や使い方もありますので，ゲームで遊ぶだけでなく，プログラムの流れや，プログラム中で各命令がどういう働きをするのかを調べてみてください。また，プログラムを改造して，あなた自身のオリジナルゲームを作ってみてください。

1 キーボード速打ちゲーム

●ゲームの遊び方

このゲームは，キーボードに慣親しむには最適のゲームといえるでしょう。画面に表示された文字とおなじキーを，ある一定時間内にキーボードから入力するゲームです。

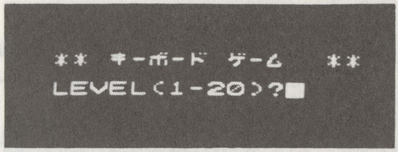
RUN RETURN または，F 8 と入力すると，画面4-2のようになります。



画面4-1

ここで，むずかしさを，1～20の値で入力します。これは，キーを押すまでの時間を決めるもので，1が一番むずかしく，20が一番やさしくなります。

数値を入力すると，A～Z，0～9の文字に囲まれた枠が表示されます。この枠の中に表示されたキーを，短時間内に入力するわけです。



画面4-2

ゲームは，「S」キーを押すと始まります。1回のゲームで，10個の文字が次々に表れますから，SCORE：に表示された数でどれだけキーボードに慣れ親しんだかがわかります。

●プログラムの改造方法

●10行，20行でセットしている文字を替えれば，カナ文字やカナグラフィックのキーボード練習になります。

現在表示される文字が36種類あるわけですが，その数を変えたい場合は，170行のRND（36）の数値を変えてください。

●レベル20でもむずかしすぎる人や，レベル1でもやさしすぎる人は，60行のN＝N＊200の数値を変えてください。むずかしすぎる人はもっと数値を大きくしてください。

●ゲーム終了後のスコアを表示するだけでなく，点数に応じて音楽などを替えたい人は，変数名Sにスコアが入っていますから，IF文を応用して，290行から300行の間にPLAY文を入れてください。

●プログラムリスト

```
10 K$="1234567890QWERTYUI"
20 K1$="OPASDFGHJKLZXCVBNM"
30 CLS:LOCATE 0,0:S=0
40 PRINT"** キーボード ゲーム **"
:PRINT
50 INPUT"LEVEL(1-20)";N
60 N=N*200:PRINT:PRINT
70 PRINT"ABCDEFGHIJKLMNO"
80 PRINT"9             P"
90 PRINT"8             Q"
```

K＄とK1＄の二つに分けて画面に表示する文字をセットします。カナ文字の練習にしたければ、ここをカナ文字に変えます（10〜20行）

枠を作成しています。カナグラフィックなどを使用すると、また違ったイメージになります（70〜90行）

```
100 PRINT"7          R"
110 PRINT"654321OZYXWVUTS"
120 PRINT:PRINT"｢S｣ｷｰｦｵｽﾄ ｽﾀ
ｰﾄ"
130 IF INKEY$<>"S" THEN 130
140 LOCATE 0,11
150 PRINT"** START ! **  "
160 FOR I=0 TO 9
170  R=RND(36)+1:J=0
180  IF R<19 THEN A$=MID$(K$
,R,1):GOTO200
190  A$=MID$(K1$,R-18,1)
200  LOCATE 7,7:PRINT A$
210  IF INKEY$=A$ THEN 250
220  J=J+1
230  IF J<N THEN 210
240  BEEP:PAUSE 5:GOTO260
250  PLAY"C7:E7:G7":S=S+1
260 NEXT
270 LOCATE 17,9:PRINT"SCORE:
";S
280 LOCATE 0,11
290 PRINT"** GAME OVER ! **"
300 LOCATE 5,15
310 INPUT"ﾓｳｲﾁﾄﾞ ｱｿﾋﾞﾏｽｶ(Y/N
)";M$
320 IF M$<>"N" THEN 30
```

（100～110行）

（150～260行）"START"の文字が画面に表示されたら、ゲームがスタートされ、10個の文字が次々と表示されます

（270～290行）スコアが表示され、ゲーム終わりのメッセージが表示されます

（300～320行）もう一度、遊ぶかどうかの判定を行っています。「N」のキーが押されると、ゲーム終了となります

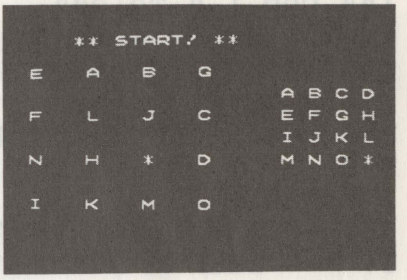
2 16パズルゲーム

●ゲームの遊び方

このゲームは，16のます目にデタラメに並んでいる15個の英文字（A～O）を，1個ずつ動かしながらアルファベット順に並べ換えるものです。図4-1の並びになれば，ゲーム終了です。このプログラムでは，英字を動かす代わりに，「*」をジョイコントローラ1の✣を使用して，上下左右に動かしてください。

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	*

図4-1



画面4-3

また，このプログラムを作るときは，最初にBGグラフィックで画面4—3のように作成してください（図4—2参照）。

●プログラムの改造方法

● BGグラフィックで書かれているAからOの文字を，数字や記号，そして15字でできている文章に換えることもできます。

●120行の最初のREMを消してしまうと，コンピュータがAからOの文字を変えるときに，いろいろな音が出るようになります。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	
0																													0
1	K70	K50	K50	K50	K20	K50	K50	K50	K20	K50	K50	K50	K20	K50	K50	K50	L00												1
2	K60				K60				K60				K60				K60												2
3	K60		A		K60		B		K60		C		K60		D		K60												3
4	K60				K60				K60				K60				K60			K70	K50	K20	K50	K20	K50	K20	K50	L00	4
5	K40	K50	K50	K50	K00	K50	K50	K50	K00	K50	K50	K50	K00	K50	K50	K50	K30			K60	A	K60	B	K60	C	K60	D	K60	5
6	K60				K60				K60				K60				K60			K40	K50	K00	K50	K00	K50	K00	K50	K30	6
7	K60		E		K60		F		K60		G		K60		H		K60			K60	E	K60	F	K60	G	K60	H	K60	7
8	K60				K60				K60				K60				K60			K40	K50	K00	K50	K00	K50	K00	K50	K30	8
9	K40	K50	K50	K50	K00	K50	K50	K50	K00	K50	K50	K50	K00	K50	K50	K50	K30			K60	I	K60	J	K60	K	K60	L	K60	9
10	K60				K60				K60				K60				K60			K40	K50	K00	K50	K00	K50	K00	K50	K30	10
11	K60		I		K60		J		K60		K		K60		L		K60			K60	M	K60	N	K60	O	K60	*	K60	11
12	K60				K60				K60				K60				K60			L10	K50	K10	K50	K10	K50	K10	K50	L20	12
13	K40	K50	K50	K50	K00	K50	K50	K50	K00	K50	K50	K50	K00	K50	K50	K50	K30												13
14	K60				K60				K60				K60				K60												14
15	K60		M		K60		N		K60		O		K60		*		K60												15
16	K60				K60				K60				K60				K60												16
17	L10	K50	K50	K50	K10	K50	K50	K50	K10	K50	K50	K50	K10	K50	K50	K50	L20												17
18																													18
19																													19
20																													20
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	

図4-2

●プログラムリスト

```
10 CLS:CGSET 1,1:VIEW
20 XA=14:YA=15:A$="*"
30 P$="CDAFGAB"
```

初期設定で、画面をクリアしたり、BGグラフィックを呼び出したりしています

```
100 FOR I=0 TO 300
110  R=RND(4)+1
120 REM P1$=MID$(D$,RND(7)+1
,1):PLAY P1$
130  ON R GOSUB 400,450,500,
550
140 NEXT
```

BGグラフィックで作成した15個の文字を1個ずつ、301回場所を入れ換えています

```
150 LOCATE 5,0:PRINT"** STAR
T! **":BEEP
160 N=0
170 D=STICK(0)
180 IF D=0 THEN 160
190 N=N+1
200 IF N<20 THEN 170
205 BEEP
210 IF D=1 THEN GOSUB400:GOT
O250
220 IF D=2 THEN GOSUB450:GOT
O250
230 IF D=4 THEN GOSUB500:GOT
O250
240 IF D=8 THEN GOSUB550
```

ジョイコントローラ1の十キーの押された位置により、各処理のサブルーチンに移行します

```
250 IF XA<>14 OR YA<>15 THEN
 160
```

終わりかどうかの判定。終わっていなければ160行に飛びます

```
260 Y=3:X=2:B$=SCR$(X,Y)
270 X=X+4:C$=SCR$(X,Y)
280 IF Y=YA AND X=XA THEN GO
TO350
290 IF B$>C$ THEN GOTO160
300 B$=C$
310 IF X<14 THEN 270
320 X=-2:Y=Y+4:GOTO270
350 LOCATE 5,0:PRINT"
     "
360 LOCATE 5,20:PRINT"** GAM
E OVER! **"
370 PLAY"T2O2:T2O2"
372 PLAY"G3GGFREG8:E3EEDRCE8
"
374 PLAY"A3FAO3C5O2A3G8:F3DF
```

終わりの処理。ゲームオーバーの表示と一緒に、音楽が演奏される

```
A5F3E8"
376 PLAY"A3FAO3CO2BA:F3DFAGF
"
378 PLAY"GAGERG:EFECRE"
380 PLAY"GGGARD:EEEFRG"
382 PLAY"O3C1O2BO3CO2B:E1FEF
"
384 PLAY"O3CO2BO3C4:EFE4:
386 PLAY"O3T4R5:O3T4R5"
390 GOTO 800
400 IF XA=14 THEN RETURN
410 XB=XA+4:GOSUB600
420 RETURN
```

ジョイコントローラ1の十キーの右側が押されたときの処理

```
450 IF XA=2 THEN RETURN
460 XB=XA-4:GOSUB600
470 RETURN
```

ジョイコントローラ1の十キーの左側が押されたときの処理

```
500 IF YA=15 THEN RETURN
510 YB=YA+4:GOSUB700
520 RETURN
```

ジョイコントローラ1の十キーの上側が押されたときの処理

```
550 IF YA=3 THEN RETURN
560 YB=YA-4:GOSUB700
570 RETURN
```

ジョイコントローラ1の十キーの上側が押されたときの処理

```
600 B$=SCR$(XB,YA)
610 LOCATE XA,YA:PRINT B$
620 LOCATE XB,YA:PRINT A$
630 XA=XB
640 RETURN
```

画面上で左右の英文字を入れ換える

```
700 B$=SCR$(XA,YB)
710 LOCATE XA,YA:PRINT B$
720 LOCATE XA,YB:PRINT A$
730 YA=YB
740 RETURN
```

画面上で上下の英文字を入れ換える

```
800 LOCATE 5,22
810 INPUT "ﾓｳｲﾁﾄﾞ ｱｿﾋﾞﾏｽｶ(Y/
N)";M$
820 IF M$<>"N" THEN RUN
```

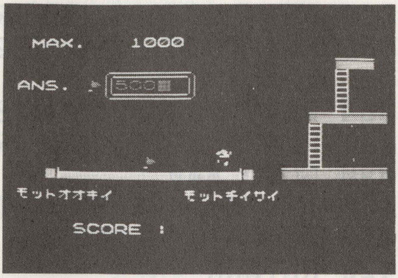
もう一度、遊ぶかどうかの判定を行っています。「N」のキーが押されると、ゲーム終了となります。

3 数あてゲーム

●ゲームの遊び方

このゲームは，名前のとおりある数を当てるゲームです。当てようとする数はコンピュータにしかわかりません。

RUN [RETURN] または，[F 8] と入力すると，画面4-4のようになります。このメニュー画面から，当てようとする数の範囲を決め，該当する数字を入力して [RETURN] キーを押すと，いよいよゲームの始まりです。これから，コンピュータの考えた数字を当てるわけです。数字を入力すると，入力した数が当てようとする数より大きいか，小さいかによって，マリオ君が動いて「もっと大きい」とか「もっと小さい」と教えてくれます。そして，みごと当てると，当てるまで回数によりマリオ君の動きが変化して，その結果を表します。当てるまでの回数が少なければ，マリオ君は一番上まで階段をかけ昇り，回数がもう少し多くなると階段の途中までしか昇りません。また，あまりにも回数が多すぎますとマリオ君は階段を落ちてしまいます。



画面4-4

また，このプログラムを作るときは，最初にBGグラフィックで，画面4-4のように作成してください（図4-3参照）。

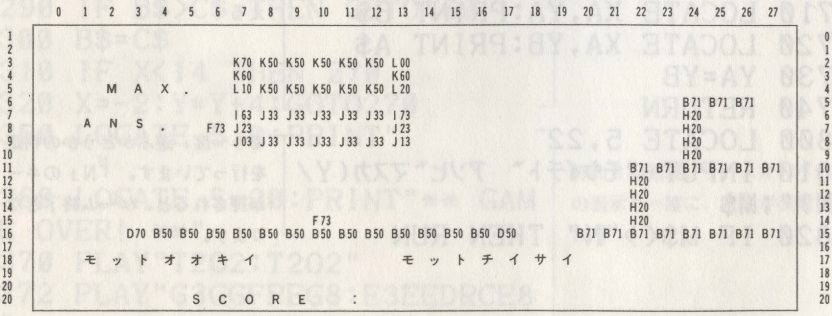


図4-3

●プログラムリスト

```
10 CLS:GOSUB 1000
15 CLS:CGSET 1,1:VIEW
20 SPRITE ON
30 S=0:X=88:Y=128
```

初期設定で、画面をクリアしたり、BGグラフィックを呼び出したりしています

```
40 GOSUB450
50 IF X=88 AND Y=128 THEN 10
0
80 LOCATE 5,22
82 PRINT"** GAME OVER! **"
83 PLAY"T1R3"
84 FOR I=0 TO 100
85  P$=MID$("CDEFGAB",RND(7)
+1,1)
86  PLAY P$
88 NEXT
90 BEEP:PAUSE 200:SPRITE OFF
92 LOCATE 5,22
94 INPUT "ﾓｳｲﾁﾄﾞ ｱｿﾋﾞﾏｽｶ(Y/N
)";M$
96 IF M$<>"N" THEN 10
98 END
```

ゲーム終了の処理ともう一度、遊ぶかどうかの判定を行っています。「N」のキーが押されると、ゲーム終了となります

```
100 BEEP
110 LOCATE 8,3:PRINT N
120 AN=RND(N+1)
130 LOCATE 8,7:INPUT"",A:S=S
+1
```

コンピュータが考えだす範囲の最大値を表示して、ある特定の数を決めます。そして、キーボードからの入力をまちます

```
140 IF A<AN THEN 500
150 IF A>AN THEN 600
```

コンピュータが考えだした数と、キーボードから入力された数が同じかどうかを判定します。数が小さいときや大きいときは、それぞれの処理のサブルーチンに移行します

```
160 PLAY"T2A5R3A4R2A1A8"
170 LOCATE 12,20:PRINT S
180 H1=0:N1=N
190 N1=N1/2:H1=H1+1
200 IF N1>0 THEN 190
210 H2=H1*2
300 XS=XE:XE=184:GOSUB800
310 YS=Y:YE=88:GOSUB900
320 IF S<H2 THEN 400
330 FOR Y=88 TO 128 STEP 4
```

コンピュータが考えだした数と、キーボードから入力された数が同じときの処理で、入力された回数でマリオ君の動きを決めます

マリオ君が階段を落ちるときの処理

```
340 DEF SPRITE0,(0,1,0,0,0)=
CHR$(24)+CHR$(25)+CHR$(26)+C
HR$(27)
350 SPRITE0,160,Y
360 NEXT:Y=Y-4:GOSUB450
370 GOTO80
```

マリオ君が階段を昇るときの処理

```
400 XS=XE:XE=200:GOSUB800
410 IF S>H1 THEN GOSUB450:GO
TO80
420 YS=Y:YE=48:X=X+8:GOSUB90
0
430 GOSUB450:GOTO80
450 DEF SPRITE0,(0,1,0,0,0)=
CHR$(4)+CHR$(5)+CHR$(6)+CHR$
(7)
460 SPRITE 0,X,Y
470 RETURN
```

マリオ君が横に動くときの処理

```
500 XS=X:XE=56:GOSUB700
510 GOSUB450
520 GOTO130
600 XS=X:XE=136:GOSUB800
610 GOSUB450
620 GOTO130
```

マリオ君が左から右に動くときのサブルーチン

```
700 P=0
710 FOR X=XS TO XE STEP -4
720 DEF SPRITE0,(0,1,0,0,0)=
CHR$(P)+CHR$(P+1)+CHR$(P+2)+
CHR$(P+3)
730 SPRITE0,X,Y
740 P=P+4:IF P>8 THEN P=0
750 PAUSE5
760 NEXT:X=X+4
770 RETURN
```

マリオ君が右から左に動くときのサブルーチン

```
800 P=0
810 FOR X=XS TO XE STEP 4
820 DEF SPRITE0,(0,1,0,1,0)=
CHR$(P+1)+CHR$(P)+CHR$(P+3)+
CHR$(P+2)
830 SPRITE0,X,Y
840 P=P+4:IF P>8 THEN P=0
```

```
850 PAUSE5
860 NEXT:X=X-4
870 RETURN
900 P=0
910 FOR Y=YS TO YE STEP -4
920 DEF SPRITE0,(0,1,0,0,0)=
CHR$(20+P)+CHR$(21-P)+CHR$(2
2+P)+CHR$(23-P)
930 SPRITE0,X,Y
940 P=P+1:IF P>1 THEN P=0
950 PAUSE5
960 NEXT:Y=Y+4
970 RETURN
1000 LOCATE 3,3
1010 PRINT "カズアテ ゲーム"
1020 LOCATE 3,10
1030 PRINT "1: 1カラ  10 ノ アイ
ダ"
1040 LOCATE 3,12
1050 PRINT "2: 1カラ  100 ノ アイ
ダ"
1060 LOCATE 3,14
1070 PRINT "3: 1カラ 1000 ノ アイ
ダ"
1080 LOCATE 3,17
1090 INPUT "";T
1100 IF T<>1 AND T<>2 AND T<
>3 THEN GOTO 1080
1110 IF T=1 THEN N=10
1120 IF T=2 THEN N=100
1130 IF T=3 THEN N=1000
1140 RETURN
```

マリオ君が階段を昇るときのサブルーチン（850〜970）

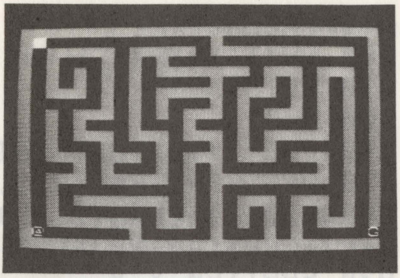
初期メニューの作成処理（1000〜1140）

4 迷路脱出ゲーム

●ゲームの遊び方

このゲームは，二人で遊ぶゲームです。画面に描かれた迷路を，ジョイコントローラで操作しながら，早くゴールにたどりついた人が勝ちになります。

遊び方は簡単で，RUN [RETURN] または，[F 8] を押すと，画面に迷路が作成され始めます。そして，最後にゴールと二人の駒が表れ，音楽が鳴ったらゲーム開始となります。



画面 4－5

ジョイコントローラの1の人が■(青い四角)，ジョイコントローラ2の人が@となり，ジョイコントローラの✜を操作して，”G”と書かれたゴールをめざすわけです。もちろん，壁を突き抜けることはできません。

●プログラムの改造方法

●301行のN＞30の数値をいろいろと変えてみると，迷路のパターンが変わってきます。数値を大きくすればするほど作成するのに時間がかかります。複雑な迷路になってきます。

```
10 GOSUB 5000
100 MX=X+(X>2)*I:MY=Y+(Y>2)*
I
105 PX=X-(X<26)*I:PY=Y-(Y<20
)*I
110 D=0
120 IF SCR$(X,MY)=" "THEN D(
D)=1:D=D+1
130 IF SCR$(PX,Y)=" "THEN D(
D)=2:D=D+1
140 IF SCR$(MX,Y)=" "THEN D(
D)=3:D=D+1
150 IF SCR$(X,PY)=" "THEN D(
D)=4:D=D+1
160 IF D=0 THEN 300
165 V=X:W=Y
170 ON D(RND(D)) GOSUB 210,2
20,230,240
```

(100～150) 壁の無い方向を調べます

(160～170) ゴールがふさがらないか調べながらメイロを作成する

```
180 LOCATE X,Y:PRINT" "
190 LOCATE V,W:PRINT" "
200 LOCATE NX,NY:PRINT""
205 X=NX:Y=NY:GOTO 100
210 NY=MY:W=NY+1:NX=X:RETURN
220 NX=PX:V=NX-1:NY=Y:RETURN
230 NX=MX:V=NX+1:NY=Y:RETURN
240 NY=PY:W=NY-1:NX=X:RETURN
300 IF FG=1 THEN RETURN
301 N=N+1:IF N>30 THEN 400
305 ON RND(7)+1 GOSUB 310,32
0,330,340,350,320,340
307 GOTO 100
310 X=0:Y=RND(9)*2:RETURN
320 X=RND(12)*2:Y=0 :RETURN
330 X=26:Y=RND(9)*2:RETURN
340 X=RND(12)*2:Y=20:RETURN
350 X=RND(12)*2:Y=RND(9)*2
360 IF SCR$(X,Y)=" " THEN RE
TURN
370 GOTO 350
400 FG=1
405 FOR K=2 TO 24 STEP 2:FOR
 L=2 TO 18 STEP 2
415 X=K:Y=L
416 IF SCR$(X,Y)=" " THEN LO
CATE K,L:PRINT"":LOCATE K,L
+1:PRINT" "
420 GOSUB 100
430 NEXT
440 NEXT
500 FG=0:N=0:X=1:Y=1:V=1:W=1
9:OX=X:OY=Y:OV=V:OW=W
510 LOCATE 26,19:PRINT"G"
520 PLAY"CDECDE"
530 LOCATE X,Y:PRINT" "
540 LOCATE V,W:PRINT"@"
600 ST=STICK(0)
610 IF ST=8 THEN Y=Y+(Y>0 AN
D SCR$(X,Y-1)=" ")
```

コントローラ1の押された位置により、駒を動かせるかどうか判断し、移動可能であれば移動する

```
620 IF ST=4 THEN Y=Y-(Y<19 A
ND SCR$(X,Y+1)=" ")
630 IF ST=2 THEN X=X+(X>0 AN
D SCR$(X-1,Y)=" ")
640 IF ST=1 THEN X=X-(X<26 A
ND SCR$(X+1,Y)=""):IF SCR$(
X+1,Y)="G" THEN 1000
650 LOCATE X,Y:PRINT" "
655 IF ST=0 THEN GOTO 700
670 LOCATE OX,OY:PRINT"
675 OX=X:OY=Y
700 ST=STICK(1)
710 IF ST=8 THEN W=W+(W>0 AN
D SCR$(V,W-1)=" ")
720 IF ST=4 THEN W=W-(W<19 A
ND SCR$(V,W+1)=" ")
730 IF ST=2 THEN V=V+(V>0 AN
D SCR$(V-1,W)=" ")
740 IF ST=1 THEN V=V-(V<26 A
ND SCR$(V+1,W)=" "):IF SCR$(
V+1,W)="G" THEN 1100
750 LOCATE V,W:PRINT "@"
755 IF ST=0 THEN 600
770 LOCATE OV,OW:PRINT " "
775 OV=V:OW=W
780 GOTO 600
```

コントローラ2の押された位置により、駒を動かせるかどうか判断し、移動可能であれば移動する

```
1000 PLAY "CDE"
1010 LOCATE 10,7
1020 PRINT "PLAYER 1 WIN!"
1030 GOTO 2000
1100 PLAY"EDC"
1110 LOCATE 10,7
1120 PRINT "PLAYER 2 WIN!"
2000 LOCATE  5,12
2010 PRINT "PRESS ANY KEY."
2020 IF INKEY$="" THEN 2020
2030 GOSUB 5010:GOTO 100
5000 DIM D(4)
5010 CLS
```

枠を作成

```
5020 PRINT "■■■■■■■■■■■■■■■■■■■■■■■■■■■"
5030 FOR J=1 TO 19
5040 LOCATE 0,J:PRINT "■                         ■"
5050 NEXT
5060 LOCATE 0,20:PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■"
5070 I=2:X=RND(11)*2
5075 Y=RND(2)*20
5080 RETURN
```

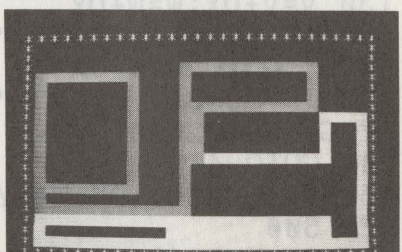
青四角（27個）（5020）
両端を青四角（スペース25個）（5040）
青四角（27個）（5060）

5 ライトサークルゲーム

●ゲームの遊び方

映画TRONでおなじみのライトサークルです。二人で遊んだり，コンピュータと対戦することもできます。

遊び方は簡単で，RUN RETURN または F8 を押すと，画面に枠が作られます。その真中にある■（青い四角）がプレイヤー1となり，□（白い四角）がプレイヤー2，もしくはコンピュータとなります。あとは，ジョイコントローラの✣を操作して，段々増えてくる壁をよけていきます。そして，先に壁にぶつかった方が負けとなります。



画面 4-6

●プログラムの改造方法

●460行のRND（30）の数値を小さくすると，コンピュータ側の動きが複雑になり，いろいろと方向を変えてきます。

```
5 DIM D(4)
6 GOSUB 900
7 PLAY"FG"
10 X=13:Y=12
20 V=14:W=Y
30 CX=-1:DX=1:CY=0:DY=CY
40 S0=STICK(0):S1=STICK(1)
42 IF FG=1 THEN 50
45 GOSUB 450
50 IF S0=0 THEN 80
60 CX=(S0=2)-(S0=1)
70 CY=(S0=8)-(S0=4)
80 IF S1=0 THEN 100
90 DX=(S1=2)-(S1=1)
92 DY=(S1=8)-(S1=4)
100 LOCATE X,Y:PRINT"■"
110 LOCATE V,W:PRINT"□"
120 X=X+CX:Y=Y+CY
130 V=V+DX:W=W+DY
135 IF SCR$(X,Y)<>" "THEN PL
AY"E":GOTO2000
136 IF SCR$(V,W)<>" "THEN PL
AY"G":GOTO2010
140 GOTO 40
450 IF SCR$(V+DX,W+DY)<>" "T
HEN 500
460 IF RND(30)=2 THEN 500
470 RETURN
500 I=0
510 IF SCR$(V+1,W)=" "THEN D
(I)=1:I=I+1
520 IF SCR$(V-1,W)=" "THEN D
(I)=2:I=I+1
530 IF SCR$(V,W+1)=" "THEN D
(I)=4:I=I+1
540 IF SCR$(V,W-1)=" "THEN D
(I)=8:I=I+1
550 S1=D(RND(I))
560 RETURN
```

100行：青四角が入ります。青四角はカナグラフィックモードにして[¥]のキーでしたね

110行：白四角が入ります。白四角はカナグラフィックモードにして[^]のキーでしたね

135行：プレイヤー1が壁にぶつかったかの判定

136行：プレイヤー2が壁にぶつかったかの判定

450〜560行：対戦相手がコンピュータのときのコンピュータの方向判断

メニュー画面の作成

```
900 CLS
910 GOSUB 1000
920 LOCATE 3,10
930 PRINT "1 PLAYER1 VS PLAY
ER2"
940 LOCATE 3,12
950 PRINT "2 PLAYER1 VS COMP
UTER"
960 N$=INKEY$
970 IF N$="" THEN 960
980 IF N$="1" THEN FG=1:GOTO
1000
990 IF N$="2" THEN FG=0:GOTO
1000
995 GOTO 960
```

初期画面作成サブルーチン

```
1000 LOCATE 0,0
1005 PRINT"******************
***********"
1010 FOR J=1 TO 20
1015 LOCATE 0,J
1020 PRINT"*
          *"
1030 NEXT
1032 LOCATE 0,21
1035 PRINT"******************
***********"
1040 RETURN
```

もう一度、遊ぶかどうかの判定を行っています。「N」のキーが押されると、ゲーム終了となります

```
2000 LOCATE 3,10:PRINT"PLAYE
R2 WIN!":GOTO 2020
2010 LOCATE 3,10:PRINT"PLAYE
R1 WIN!"
2020 LOCATE 3,11:PRINT"モウイチト
゛ ?(Y/N)"
2030 Y$=INDEY$
2040 IF Y$="" THEN 2030
2050 IF Y$="N" THEN STOP
2060 GOTO 6
```

㈱マイクロデザインでは，皆さんのオリジナルな作品を募集しています。独創的で楽しい，あなたの個性あふれるプログラムをどしどしカセットに入れてお送りください。優秀な作品は，ファミコンファンクラブの「ファミコンジャンプ」に掲載させていただきます。

送り先　㈱マイクロデザイン
東京都中央区築地３－２－３　シルバールームズ1004号　(〒104)
ＴＥＬ　(03)543－1601

●本書の内容に関するお問い合わせ

電話によるお問い合わせは，日・祝日を除く，午前10時〜午後６時までとさせていただきます。

手紙によるお問い合わせは，必ず往復ハガキか，返信用封筒を同封してください。

㈱マイクロデザイン
東京都中央区築地３－２－３　シルバールームズ1004号　(〒104)
ＴＥＬ　(03)543－1601

ファミリーベーシックで遊ぼう！
ぼくとわたしの
ファミリーベーシック教室

1985年４月11日　初　版 第１刷 発行

著　　者　㈱マイクロデザイン
発 行 人　武内　静夫
発 行 所　㈱マイクロデザイン
東京都中央区築地３－２－３　(〒104)
ＴＥＬ　(03)543－1601
販 売 元　㈱マイクロハウス
東京都中央区新富１－３－９　(〒104)
ＴＥＬ　(03)552－1530
印刷・製本　㈱そうご

ファミコンファンクラブ会員募集要項

（株）マイクロデザインでは，ファミコンクラブの会員を募集いたしております。

ファミコンクラブに入会されますと，次のような特典がございます。

【特　典】

1．全員に会員証と会員バッジをお送りします。
2．定期的に『ファミコンジャンプ』を無料でお送りします。
3．随時，ファミリーコンピュータまたはファミリーベーシックの講習会・セミナーのご案内をさしあげます。
4．その他，たのしい企画・特典を多数用意しています。

【会員規則】

1．会員はファミリーコンピュータを持っていること。
2．会員は，ファミリーコンピュータのゲームで遊ぶのが好きであること。

……………………………………キリトリ線……………………………………

ファミコンファンクラブ入会申込書

名前　　　　　　（　　）才	男・女	TEL（　　）
住所　〒		
ファミリーベーシックを持っている	はい	いいえ
ファミリーベーシックV3を持っている	はい	いいえ

キリトリ線

3．入会金は，無料です。

4．年会費は，2千円とします。

【応募要領】

入会ご希望の方は，下段の申し込み用紙に必要事項を記入の上，年会費として2000円分の定額郵便為替（無記入・郵便局で扱っています）を同封していただくか，または現金書留にて下記の住所までお送りください。お申し込みには，下段の申し込み用紙以外は，使えませんからご注意ください。

なお，小・中学生のかたは，保護者の承諾を得てください。

【送り先】

〒104　東京都中央区築地3－2－3シルバールームズ1004号
（株）マイクロデザイン　ファミコンファンクラブ係

……………………………………キリトリ線……………………………………

お手持ちのゲームカセットに○を付けてください

・ドンキーコング　・ゴルフ　・五目ならべ　・ベースボール
・ドンキーコングＪＲ　・ポパイ　・ワイルドガンマン　・麻雀
・ドンキーコング3　・ピンボール　・ダックハント　・四人打ち麻雀
・マリオブラザーズ　・テニス　・ホーガンズアレイ　・エキサイトバイク
・ロードランナー　・Ｆ1レース　・デビルワールド　・アイスクライマー
・ナッツ＆ミルク　・ゼビウス　・クルクルランド　・バルーンファイト
・アーバンチャンピオン　・ＤＫＪＲの算数遊び　・ポパイの英語遊び
・ギャラクシアン　・マッピー　・パックマン　・その他（　　　　　　）

つぎに㊙攻略法シリーズで読みたいゲームをお書きください。

キリトリ線

MICRO DESIGN

¥980

ファミリーベーシックで遊ぼう!

ぼくとわたしのファミリーベーシック教室

マイクロデザイン編